



## Moving Horizon Estimation and Control

**Jørgensen, John Bagterp**

*Publication date:*  
2004

*Document Version*  
Publisher's PDF, also known as Version of record

[Link back to DTU Orbit](#)

*Citation (APA):*  
Jørgensen, J. B. (2004). *Moving Horizon Estimation and Control*. Technical University of Denmark.

---

### General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

---

# Moving Horizon Estimation and Control

---

John Bagterp Jørgensen

December 2004

Department of Chemical Engineering  
Technical University of Denmark

Copyright © John Bagterp Jørgensen, December 2004  
ISBN XX-XXXXXX-XX-X  
Printed by Book Partner, Nørhaven Digital, Copenhagen, Denmark

# Preface

This dissertation is submitted to the Technical University of Denmark in partial fulfillment of the requirements for the degree of Doctor of Philosophy (Ph.D.) in chemical engineering.

The research reported in this dissertation is conducted partly at Department of Chemical Engineering at the Technical University of Denmark with professor Dr. Sten Bay Jørgensen as supervisor and partly at Department of Chemical Engineering at the University of Wisconsin-Madison with professor Dr. James B. Rawlings as supervisor. Professor Dr. Sten Bay Jørgensen and professor Dr. James B. Rawlings have contributed to the ideas, technologies, and results reported in this dissertation. However, any point of views expressed are not necessarily the point of view of professor Dr. Sten Bay Jørgensen, professor Dr. James B. Rawlings or the institutions to which they are affiliated. Furthermore, any shortcomings are alone my responsibility.

The study and reporting is conducted in the period august 1997 to december 2002. Final adjustments of the thesis was conducted december 2004. The period august 1997 to july 2000 was financed by a DTU scholarship (Ph.D. no. 97-0076-011). The period august 2000 to january 2001 was financed by employment as assistant research professor and research assistant, respectively, as compensation for extraordinary teaching activities during the ordinary Ph.D. study period. During the period march to july 2001, I was employed at 2-control ApS and financed by this company. The research in august and september 2001 was financed by employment at the OPERA project as research assistant. The remaining time has been financed by John Bagterp Jørgensen. I am grateful for the financial support and acknowledge the institutions, companies, agencies and private persons for the financial support of this research.

John Bagterp Jørgensen

Malmö, December 2004



# Summary

This dissertation concerns numerical procedures for the problems arising in moving horizon estimation and control. Moving horizon estimation and control is also referred to as model predictive control as well as receding horizon estimation and control. Model predictive control is the most successful and applied methodology beyond PID-control for control of industrial processes. The main contribution of this thesis is introduction and definition of the *extended linear quadratic optimal control problem* for solution of numerical problems arising in moving horizon estimation and control. An efficient structure-employing methodology for solution of the extended linear quadratic optimal control problem is provided and it is discussed how this solution is employed in solution of constrained model predictive control problems as well as in the solution of nonlinear optimal control and estimation problems.

**Chapter 1** motivates moving horizon estimation and control as a paradigm for control of industrial processes. It introduces the *extended linear quadratic control problem* and discusses its central role in moving horizon estimation and control. Introduction, application and efficient solution of the extended linear quadratic control problem is the key contribution of this thesis. In addition chapter 1 provides a comprehensive survey of existing methods for model predictive control.

**Chapter 2** discusses computational methods and inherent approximations in model predictive control. By considering the stochastic optimal control problem, the approximations and assumptions of model predictive control are pinpointed. In an ad hoc fashion the separation principle and certainty-equivalence are assumed to prevail, such that the stochastic optimal control problem may be separated into an estimation problem and a deterministic optimal control problem. Both the estimation problem and the obtained deterministic optimal control problem are demonstrated to be instances of a constrained nonlinear optimal control problem. In the sequential quadratic programming algorithm for solution of constrained nonlinear optimal control problems, the quadratic subproblem generated at each iteration is shown to be a constrained linear-quadratic optimal control problem. Procedures for generation of the constrained linear-quadratic optimal control problem and its data from the nonlinear estimation problem, the nonlinear control problem, the linear moving horizon estimator, and the linear moving horizon controller are provided. The significance of these conversions to constrained linear-quadratic optimal control problems is that the entire model predictive control problem can be solved efficiently by having efficient algorithms tailored for solution of the con-

strained linear-quadratic optimal control problem. The major intention in this chapter is to emphasize the central role of the constrained linear-quadratic optimal control problem in model predictive control such that tailored algorithms for the constrained linear-quadratic optimal control problem is motivated and justified.

**Chapter 3.** A primal active set, a dual active set, and an interior point algorithm for solution of the constrained linear quadratic optimal control problem are outlined. The major computational effort in all these algorithms reduces to solution of certain unconstrained linear quadratic optimal control problems, i.e. the extended linear quadratic control problem. A Riccati recursion procedure for effective solution of such unconstrained problems is stated.

**Chapter 4.** Based on dynamic programming, Riccati recursion procedures for the linear-quadratic optimal control problem as well as the extended linear-quadratic optimal control problem are developed. Compared to alternative solution procedures such as control vector parameterization by elimination of the states, the Riccati based procedure is highly efficient for long prediction horizons. The extended linear-quadratic optimal control problem may also be regarded as an equality constrained quadratic program with special structure. The computation of the optimal solution-Lagrange multiplier pair for a convex equality constrained quadratic program is specialized to the extended linear-quadratic optimal control problem treated as a quadratic program. Efficient solution of the highly structured KKT-system corresponding to the extended linear-quadratic optimal control problem is facilitated by the Riccati recursion developed by dynamic programming.

**Chapter 5** presents the principles for efficient solution of unconstrained nonlinear optimal control problems described by ordinary differential equations. These principles are presented through numerical solution of a continuous-time nonlinear optimal control problem of the Bolza form. To focus on the basic principles involved and for illustrative purposes, the continuous-time Bolza problem is discretized by the explicit Euler method. The discrete-time nonlinear optimal control problem of the Bolza form is solved by different SQP methods and an algorithm based on the discrete maximum principle. The SQP algorithms presented are implementations based on open- and closed-loop feasible path control vector parameterizations as well as an infeasible path simultaneous procedure. Two procedures for solution of the quadratic programs are presented. In the first procedure, the structure of the quadratic programs arising in the solution of the nonlinear optimal control problem is utilized by a Riccati iteration based factorization of the resulting KKT-system. In the second procedure, an efficient procedure for elimination of the states and solution of a dense reduced space quadratic program is presented. These methods are compared for a simple process example operated around an unstable equilibrium. The infeasible path and the closed-loop feasible path algorithms converge for this example. The implemented open-loop feasible path algorithms are not able to converge to an unstable equilibrium. The Riccati based solution procedure

enables implementation of the stabilized infeasible path SQP algorithm as well as the closed-loop feasible path SQP algorithm. The methods are presented in a framework that is easily extended to constrained nonlinear optimal control problems. Such extensions and methodologies for efficient integration of the ordinary differential equations as well as the corresponding sensitivity equations are discussed.

**Chapter 6** summarizes the main contribution of this thesis. It briefly discusses the pros and cons of using the extended linear quadratic control framework for solution of deterministic optimal control problems.

**Appendices.** Appendix A demonstrates how quadratic programs arise in sequential quadratic programming algorithms. Appendix B uses a control vector parameterization approach to express various extended constrained linear quadratic optimal control problems as standard quadratic programs. Appendix C discuss construction of maximal output admissible sets. It provides an algorithm for computation of the maximal output admissible set for linear model predictive control. Appendix D provides results concerning linear regression. Appendix E discuss prediction error methods for identification of linear models tailored for model predictive control.





# Dansk Resumé

Denne afhandling omhandler numeriske metoder til løsning af de opgaver der opstår i rullende horisonts estimering og styring. Rullende horisonts estimering og styring kaldes også model prædiktiv regulering. Udover PID-regulering er model prædiktiv regulering den mest succesfulde og anvendte reguleringsteknologi til styring af industrielle processer. Hovedbidraget i denne afhandling er indførsel og definition af det *udvidede lineære kvadratiske styrings problem* til løsning af numeriske opgaver, der opstår indenfor rullende horisonts estimering og styring. En effektiv struktur-udnyttende metode til løsning af det udvidede lineære kvadratiske styrings problem udledes, og det diskuteres, hvordan denne løsning anvendes til såvel løsning af model prædiktive reguleringsopgaver med begrænsninger som ulineære optimale styrings og estimeringsopgaver. Den effektive struktur-udnyttende løsningsmetode er baseret på Riccati iterationer.



# Contents

|   |            |
|---|------------|
| <b>Preface</b>  | <b>iii</b> |
| <b>Summary</b>  | <b>v</b>   |
| <b>Dansk Resumé</b>   | <b>ix</b>  |
| <b>1 Introduction</b>   | <b>1</b>   |
| 1.1 Models for Predictive Control . . . . .                       | 6          |
| 1.1.1 Linear Time Invariant Model Representations . . . . .       | 6          |
| 1.1.2 Process Control Literature . . . . .                        | 8          |
| 1.1.3 Generalized Predictive Control . . . . .                    | 13         |
| 1.1.4 Discrete-Time State Space Models and Kalman . . . . .       | 17         |
| 1.1.5 Continuous-Time State Space Models . . . . .                | 27         |
| 1.1.6 Neural Network Models . . . . .                             | 28         |
| 1.2 Prediction Horizons and Stability . . . . .                   | 30         |
| 1.2.1 Finite-Horizon Optimal Regulation and Control . . . . .     | 31         |
| 1.2.2 Infinite-Horizon Optimal Regulation . . . . .               | 31         |
| 1.2.3 Transcription to the Regulation Problem . . . . .           | 42         |
| 1.2.4 Infinite-Horizon Optimal Control . . . . .                  | 46         |
| 1.3 Constraints . . . . .   | 50         |
| 1.3.1 Hard Constraints . . . . .                                  | 50         |
| 1.3.2 Soft Constraints Motivated by Infeasibilities . . . . .     | 52         |
| 1.4 The Control Problem . . . . .                                 | 56         |
| 1.5 Thesis Organization . . . . .                                 | 63         |
| <b>2 The Computational Principles of Model Predictive Control</b> | <b>67</b>  |
| 2.1 Introduction . . . . .  | 68         |
| 2.1.1 Organization and Purpose . . . . .                          | 68         |
| 2.1.2 Literature Review . . . . .                                 | 70         |
| 2.2 Model Predictive Control . . . . .                            | 76         |

---

|          |  |            |
|----------|--|------------|
| 2.2.1    | Stochastic Optimal Control . . . . .                               | 77         |
| 2.2.2    | Suboptimal Controllers . . . . .                                   | 79         |
| 2.2.3    | State Estimation . . . . .   | 81         |
| 2.2.4    | Moving Horizon Estimation and Control . . . . .                    | 84         |
| 2.2.5    | Stochastic Optimal Control of Uncertain Systems . . . . .          | 87         |
| 2.3      | Linear Quadratic Optimal Control . . . . .                         | 91         |
| 2.3.1    | Numerical Methods . . . . .  | 93         |
| 2.4      | Nonlinear Optimal Control . . . . .                                | 96         |
| 2.4.1    | BFGS Approximation . . . . .                                       | 102        |
| 2.4.2    | Least Squares Objective . . . . .                                  | 106        |
| 2.5      | Nonlinear Moving Horizon Estimation . . . . .                      | 111        |
| 2.5.1    | Basic Nonlinear Moving Horizon Estimator . . . . .                 | 112        |
| 2.5.2    | Offset Free Nonlinear Moving Horizon Estimator . . . . .           | 117        |
| 2.5.3    | Moving Horizon Estimation . . . . .                                | 121        |
| 2.6      | Moving Horizon Control . . . . .                                   | 131        |
| 2.6.1    | Nonlinear Moving Horizon Control . . . . .                         | 131        |
| 2.6.2    | Least-Squares Nonlinear Moving Horizon Control . . . . .           | 135        |
| 2.6.3    | Moving Horizon Control with Linear Models . . . . .                | 144        |
| 2.6.4    | Anticipatory Control . . . . .                                     | 151        |
| 2.7      | Conclusion . . . . .   | 153        |
| <b>3</b> | <b>Numerical Methods for Moving Horizon Estimation and Control</b> | <b>155</b> |
| 3.1      | Introduction . . . . .   | 155        |
| 3.2      | Algorithms for Solution of Convex QPs . . . . .                    | 157        |
| 3.3      | KKT-Systems . . . . .  | 159        |
| 3.4      | Linear Quadratic Optimal Control . . . . .                         | 161        |
| 3.5      | Model Predictive Control . . . . .                                 | 164        |
| 3.6      | Moving Horizon Estimation . . . . .                                | 166        |
| 3.7      | Conclusion . . . . .   | 168        |
| <b>4</b> | <b>Dynamic Programming for Linear-Quadratic Optimal Control</b>    | <b>169</b> |
| 4.1      | Dynamic Programming . . . . .                                      | 170        |
| 4.2      | Standard Linear-Quadratic Optimal Control . . . . .                | 173        |

---

|          |   |            |
|----------|---|------------|
| 4.3      | Extended Linear-Quadratic Optimal Control . . . . .             | 178        |
| 4.3.1    | Algorithms . . . . .  | 185        |
| 4.4      | Equality Constrained Quadratic Program . . . . .                | 188        |
| 4.4.1    | Extended Linear-Quadratic Optimal Control Problem . . . . .     | 192        |
| 4.5      | Conclusion . . . . .  | 201        |
| <b>5</b> | <b>Numerical Solution of Nonlinear Optimal Control Problems</b> | <b>203</b> |
| 5.1      | Introduction . . . . .  | 204        |
| 5.2      | Sequential Quadratic Programming . . . . .                      | 205        |
| 5.2.1    | Optimality Conditions . . . . .                                 | 207        |
| 5.2.2    | Newton's Method and SQP . . . . .                               | 210        |
| 5.2.3    | Merit Function and Step Length . . . . .                        | 214        |
| 5.3      | SQP for Nonlinear Optimal Control . . . . .                     | 218        |
| 5.3.1    | Solution Methods . . . . .                                      | 218        |
| 5.3.2    | Transcription in the Direct Methods . . . . .                   | 221        |
| 5.3.3    | The Lagrangian and Partial Separability . . . . .               | 222        |
| 5.3.4    | Partitioned BFGS Hessian Approximation . . . . .                | 223        |
| 5.3.5    | The Quadratic Program . . . . .                                 | 225        |
| 5.3.6    | Line search and Penalty Functions . . . . .                     | 230        |
| 5.3.7    | The SQP Algorithm for Optimal Control . . . . .                 | 232        |
| 5.3.8    | Discretization . . . . .  | 234        |
| 5.4      | Feasible Path SQP . . . . .                                     | 235        |
| 5.4.1    | Open-Loop Feasible Path SQP . . . . .                           | 238        |
| 5.4.2    | Closed-Loop Feasible Path SQP . . . . .                         | 240        |
| 5.5      | Reduced-Space Quadratic Program . . . . .                       | 244        |
| 5.5.1    | Dense Quadratic Program . . . . .                               | 244        |
| 5.5.2    | Direct Formation of the Dense Quadratic Program . . . . .       | 248        |
| 5.6      | Discrete Maximum Principle . . . . .                            | 252        |
| 5.7      | Process Example . . . . .                                       | 255        |
| 5.8      | Extensions and Refinements . . . . .                            | 260        |
| 5.8.1    | Inequality Constraints . . . . .                                | 261        |
| 5.8.2    | Merit Function . . . . .  | 264        |
| 5.8.3    | Hessian Approximations . . . . .                                | 265        |
| 5.8.4    | Integration Methods . . . . .                                   | 270        |

|                       |   |            |
|-----------------------|---|------------|
| 5.8.5                 | Index One DAEs . . . . .  | 272        |
| 5.8.6                 | Parallelization . . . . .   | 273        |
| 5.9                   | Conclusion . . . . .  | 274        |
| <b>6</b>              | <b>Conclusion</b>   | <b>275</b> |
| <br><b>Appendices</b> |   |            |
| <b>A</b>              | <b>Quadratic Program in SQP</b>   | <b>279</b> |
| A.1                   | Quadratic Program in SQP . . . . .  | 279        |
| A.1.1                 | Exact Hessian of Lagrangian . . . . .   | 281        |
| A.1.2                 | Secant Approximation of the Hessian . . . . .                                 | 281        |
| A.1.3                 | Gauss-Newton Approximation in Least-Squares Problems                          | 282        |
| A.1.4                 | Hybrid Methods for Least-Squares Problems . . . . .                           | 284        |
| <b>B</b>              | <b>Dense MPC Quadratic Program</b>  | <b>285</b> |
| B.1                   | Standard Quadratic Program . . . . .  | 285        |
| B.2                   | LTI MPC Quadratic Program . . . . .   | 287        |
| B.3                   | MPC with Nonzero Exogenous Input . . . . .                                    | 299        |
| B.4                   | MPC with Time Variant Linear Models . . . . .                                 | 307        |
| <b>C</b>              | <b>Maximal Output Admissible Set</b>  | <b>321</b> |
| C.1                   | Motivation for Maximal Output Admissible Sets . . . . .                       | 321        |
| C.2                   | Definitions and Basic Properties . . . . .                                    | 322        |
| C.3                   | Algorithmic Determination . . . . .   | 328        |
| C.3.1                 | Algorithm . . . . .   | 334        |
| C.4                   | Examples . . . . .  | 335        |
| C.4.1                 | Bounded Observable Output Constraints with Origo in<br>the Interior . . . . . | 335        |
| C.4.2                 | Partly Unobservable Output Constraints . . . . .                              | 340        |
| C.4.3                 | Origo on the Boundary of The Constraint Set . . . . .                         | 343        |
| C.5                   | Conclusion . . . . .  | 348        |
| <b>D</b>              | <b>Linear Regression</b>  | <b>351</b> |
| D.1                   | Regression Framework . . . . .  | 351        |

---

|          |  |            |
|----------|--|------------|
| D.1.1    | Numerical Methods . . . . .                                  | 357        |
| D.2      | Univariate Regression . . . . .                              | 365        |
| D.2.1    | Least Squares Estimate . . . . .                             | 366        |
| D.2.2    | Weighted Least Squares Estimate . . . . .                    | 366        |
| D.2.3    | Maximum Likelihood Estimate . . . . .                        | 367        |
| D.2.4    | Inference . . . . .  | 369        |
| D.2.5    | Maximum a Posteriori Estimation . . . . .                    | 379        |
| D.3      | Multivariate Regression . . . . .                            | 383        |
| D.3.1    | Least Squares Regression . . . . .                           | 384        |
| D.3.2    | Maximum Likelihood Regression . . . . .                      | 387        |
| <b>E</b> | <b>Prediction-Error-Methods for Model Predictive Control</b> | <b>391</b> |
| E.1      | Introduction . . . . .                                       | 391        |
| E.2      | Predictive Control . . . . .                                 | 393        |
| E.2.1    | Filter and Predictor . . . . .                               | 394        |
| E.2.2    | Regulator . . . . .  | 395        |
| E.3      | Prediction-Error-Methods . . . . .                           | 397        |
| E.3.1    | Standard Regression Problem . . . . .                        | 397        |
| E.3.2    | The PE Method as a Regression Problem . . . . .              | 398        |
| E.3.3    | Single-Step j-Step-Ahead Prediction Error . . . . .          | 398        |
| E.3.4    | Multi-Step Prediction Error . . . . .                        | 400        |
| E.3.5    | Multi-Step Maximum Likelihood Predictors . . . . .           | 402        |
| E.4      | Continuous-Time Stochastic Model . . . . .                   | 407        |
| E.5      | SISO Example . . . . .                                       | 409        |
| E.5.1    | Identical Model and System Structure . . . . .               | 409        |
| E.5.2    | Simplified Model with Output Integrator . . . . .            | 419        |
| E.6      | Wood and Berry Distillation Example . . . . .                | 421        |
| E.6.1    | Identification of a Control Relevant Model . . . . .         | 422        |
| E.6.2    | Model Predictive Control . . . . .                           | 425        |
| E.7      | Conclusion . . . . .   | 427        |
| E.8      | Array Algorithms . . . . .                                   | 428        |
| E.8.1    | One-Step Predictions . . . . .                               | 428        |
| E.8.2    | j-Step Predictions . . . . .                                 | 429        |



|                   |            |
|-------------------|------------|
| <b>References</b> | <b>431</b> |
| <b>Vitae</b>      | <b>463</b> |

# Introduction

Faced with the increased global competition, cost competitiveness of the process industries is not just a mean to achieve a competitive advantage, it is a prerequisite for survival and profitability. In spite of the market pressure to reduce costs of operation, the market simultaneously seems to demand products of improved quality as well. This double quest for better products at lower costs has forced the process industries to reduce the variability of key product quality parameters while simultaneously minimizing the energy and raw materials needed for the production and maximizing asset utilization. Some ways to increase the asset utilization are to reduce start-up, shut-down, and product grade transition times. Other ways include increasing the throughput as a consequence of less recycle and tighter product quality control.

Dynamic optimization implemented in a moving horizon manner has successfully been applied to a number of industrial processes in order to realize the ambitions of lowering production costs, increasing asset utilization, and improving product quality by reducing the variability of key process quality indicators. The ability of dynamic optimization to directly link a company's business and economic objectives to its operations has made dynamic optimization one of the fastest growing technologies in the automation industry. Dynamic optimization is one of the most efficient ways to achieve optimal asset utilization and performance, as it is often aimed directly at improving plant profitability in an immediately quantifiable way. Almost all processes can benefit from online dynamic optimization in some way - i.e. by increased yield and throughput, limited off-spec production, reduced down-time, and lowered energy costs. In mathematical terms, dynamic optimization is referred to as a deterministic open-loop optimal control problem.

The benefits of dynamic optimization are typically realized by on-line moving horizon implementation of dynamic optimization. A moving horizon implementation means that the dynamic optimization calculations are repeated and conducted on-line each time new information such as process measurements become available. At each sample, the dynamic optimization considers a fixed window of past measurements to estimate the current state of the system. This estimated state is used along with the model to forecast the process behavior for a fixed time-window into the future. The dynamic optimization computes

the optimal sequence of manipulable variables such that the predicted process behavior is as desirable as possible subject to the physical and operational constraints of the system. Only the first element in the sequence of optimal manipulable variables is implemented on the process. As new information becomes available the horizons of the estimator and the regulator are shifted one sample forward and the procedure is repeated. This control methodology is referred to as model predictive control. Due to the way the prediction horizon is shifted by the controller, this methodology is also referred to as moving horizon control or receding horizon control. The corresponding estimation procedure is called moving horizon estimation or receding horizon estimation.

The essential feature of model predictive control, is that the control problem is formulated as a dynamic optimization problem which is repeatedly solved online. While this is by itself hardly a new idea, it constitutes one of the first examples of large-scale dynamic optimization applied routinely in real-time in the process industries. The optimization formulation allows for direct and systematic incorporation of inequality constraints in the control formulation. The constraint handling ability of model predictive controllers was one of the main reasons for their introduction in the oil industry in the 1970s. The economic optimal point of operation is often located at the intersection of constraints. In the face of disturbances and inaccurate models, operation at the mathematical optimum will lead to frequent violation of the constraints. To avoid constant violation of these constraints with the possible associated emergency shut down of facilities, the targeted operating point back-offs from the constraints (Loeblein and Perkins, 1999a). The less back-off required the smaller the economic penalty associated with not being able to operate at the mathematical optimum. To reduce the back-off and associated economic penalty with not being able to operate at the mathematical optimum, it is essential that the controller is able to handle constraints. Before the advent of model predictive control, constraints were handled by augmenting single loop controllers by various selectors and overrides (Shinskey, 1988; Seborg *et al.*, 1989). Model predictive control technology clearly outperformed these technologies as it in addition to handle constraints in a systematic and transparent manner was directly applicable to multi-variable, time-delay, and inverse response systems. Despite these additional benefits, a contributor to the industrial success of model predictive control is undoubtedly, that model predictive control techniques provide the only methodology to handle constraints in a systematic way during the design and implementation of the controller.

Being model based is both the strength and the weakness of model predictive controllers. The strength of being model based is that the model allows the controller to compensate for time-delays and interactions. The weakness is that a model is needed. Provided, the computational infrastructure for model predictive control is in place, identification of the model is often the major task in a model predictive control project. For industrial processes, it is often the case, that the physical phenomena occurring in the process are not even

known qualitatively and certainly not known quantitatively. This implies that physically based models can only be established at high costs. For processes adequately described by linear models, empirical models can be identified. The systematic methodologies for obtaining empirical linear models have had an enormous impact on the wide-spread application of linear model predictive control. The identification of nonlinear empirical models is much less well-developed and perhaps one of the major bottlenecks for wide-spread application of nonlinear model predictive control.

By being an optimization based controller, model predictive control can account for constraints in an optimal way as well as minimize some objective. The objective may be based on economic considerations, statistical considerations, or being an ad hoc expression constructed to penalize deviations from a nominal trajectory.

Global competition and liberalization of markets has led to a demand for technologies such as model predictive control that can contribute to efficient, consistent and agile manufacturing. On the supply side, the two main driving forces in advancing model predictive control technology are on the one side algorithms and computing technology for dynamic optimization and on the other side modeling capabilities. General purpose modeling software such as gProms facilitates creation of dynamic models based on physical principles. The technology for solution of dynamic optimization is required for estimation of parameters as well as state estimation and computation of the optimal control sequence. For nonlinear model predictive control, the major computational tasks in dynamic optimization are numerical solution of the ordinary differential or differential algebraic equations constituting the model along with their sensitivities and an optimization. The optimization computes the best trajectory subject to the physical and operational constraints. The optimization is typically accomplished using a sequential quadratic programming (SQP) algorithm. Besides evaluation of the model functions, the main computational effort of sequential quadratic programming concerns solution of the quadratic programs generated. For linear model predictive control, the computational effort is exclusively associated with the solution of a single quadratic program. The need for solution of a quadratic program does not only arise in the solution of the nonlinear and linear moving horizon control problem, but does also arise in the solution of the nonlinear and linear moving estimation problem as well as in the solution of the parameter estimation problem.

The quadratic programs arising for constrained moving horizon estimation and control comprise a constrained linear-quadratic optimal control problem. The constrained linear-quadratic optimal control problem is a quadratic program with special structure. This structure can be utilized in algorithms tailored for the constrained linear-quadratic optimal control problem. Utilization of the structure implies that tailored algorithms for solution of constrained linear-quadratic optimal control problem are much faster than general purpose QP solvers applied to the constrained linear-quadratic optimal control problem.

The tailored algorithms are based on solution of the first order necessary and sufficient optimality conditions by Riccati iteration. The Riccati iteration based QP solver for constrained linear-quadratic optimal control problems is particularly efficient for problems with long control horizons, whereas techniques based on state elimination and solution of the resulting dense QP are more efficient for problems with short horizons. From a stability and robustness point of view, the application of a long control horizon in model predictive control is desirable as the deviation between the computed open loop response and the nominal closed loop response decreases by the length of the control horizon. As the open loop response predicted by the controller approaches the resulting closed loop response as a consequence of long control- and prediction horizons, the tuning of the controller becomes intuitive. In contrast, short horizons with significant deviation between the open loop predictions and the resulting closed loop trajectory may lead to counter intuitive tuning and performance may deteriorate. These properties of model predictive controllers necessitates efficient algorithms for solution of long-horizon constrained linear-quadratic optimal control problems.

As has recently been explained independently by Goodwin (2002) and Binder (2002), inverse problems are germane to almost all algorithmic problems in signal processing, telecommunications, and control. Inverse problems tend to be difficult to solve, when they are ill-conditioned or when the inverse must satisfy constraints. Ill-conditioned problems are typically solved using regularization (Hansen, 1997). Constraints are added to the inverse problems for stability and robustness reasons as well as to provide an accurate description of the physical system modeled. Constrained inverse problems with linear models are quadratic programs. In estimation and control of linear dynamic systems, the quadratic programs that constitute the inverse problems are both instances of linear-quadratic optimal control problems. Linear-quadratic optimal control problem may be solved efficiently using Riccati-iteration based procedures. The analogue of constrained inverse problem for linear dynamic systems is a constrained linear-quadratic optimal control problem. Similarly, the analogue of an unconstrained inverse problem is a linear-quadratic optimal control problem. These observations emphasize that the linear-quadratic optimal control problem and the constrained linear-quadratic optimal control problem are germane in algorithmic control and estimation problems.

The linear-quadratic optimal control problem is an extension of the LQ-problem typically considered in the control systems literature (Kwakernaak and Sivan, 1972; Rugh, 1996; Franklin *et al.*, 1998). The classic time invariant LQ-problem is

$$\min_{\{\delta x_{k+1}, \delta u_k\}_{k=0}^{N-1}} \phi = \frac{1}{2} \left( \sum_{k=0}^{N-1} \delta x'_k Q \delta x_k + \delta u'_k R \delta u_k \right) + \frac{1}{2} \delta x'_N P_N \delta x_N \quad (1.1a)$$

$$s.t. \quad \delta x_{k+1} = A \delta x_k + B \delta u_k \quad k = 0, 1, \dots, N-1 \quad (1.1b)$$

in which  $\delta x_k$  and  $\delta u_k$  are deviation variables

$$\delta x_k = x_k - x_s \quad (1.2a)$$

$$\delta u_k = u_k - u_s \quad (1.2b)$$

around some steady state  $(x_s, u_s)$ . The classic time-invariant LQ-problem has an objective function in which the stage costs contain quadratic terms only. The stage costs do not contain quadratic cross terms even though these could be included. The state transition equation of the classic time-invariant LQ-problem is linear in the deviation variables. All matrices in the classic time-invariant LQ problem are invariant of time. The linear-quadratic optimal control problem is an extension of the classic time-invariant LQ-problem. Therefore, it is occasionally called the extended linear-quadratic optimal control problem. The linear-quadratic optimal control problem is time variant and not necessarily formulated in deviation variables. The linear-quadratic optimal control problem is

$$\min_{\{x_{k+1}, u_k\}_{k=0}^{N-1}} \phi = \sum_{k=0}^{N-1} l_k(x_k, u_k) + l_N(x_N) \quad (1.3a)$$

$$s.t. \quad x_{k+1} = A'_k x_k + B'_k u_k + b_k \quad k = 0, 1, \dots, N-1 \quad (1.3b)$$

in which the stage costs

$$l_k(x_k, u_k) = \frac{1}{2} (x'_k Q_k x_k + 2x'_k M_k u_k + u'_k R_k u_k) + q'_k x_k + r'_k u_k + f_k \quad (1.4a)$$

$$l_N(x_N) = \frac{1}{2} x'_N P_N x_N + p'_N x_N + \gamma_N \quad (1.4b)$$

have quadratic terms including quadratic cross terms as well as linear terms. The state transition equations (1.3b) in the linear-quadratic optimal control problem are affine rather than linear in the decision variables.

The unconstrained linear-quadratic optimal control problem and the LQ-problem are equivalent. The solution of an unconstrained linear-quadratic optimal control problem in isolation is often obtained most efficiently by solution of the equivalent LQ problem. However, the solution of nonlinear moving horizon estimation and control problems is facilitated by the linear-quadratic optimal control problem (1.3). The linear-quadratic optimal control problem (1.3) turns out to be a surprisingly effective framework for analysis and formulation of moving horizon estimation and control problems. The solution of linear model predictive control problems with pre-specified time variant reference and load trajectories is facilitated by the linear-quadratic optimal control formulation.

The main contribution of this thesis is to demonstrate the usefulness and application of the extended linear quadratic optimal control problem (1.3).

## 1.1 Models for Predictive Control

The model used for the predictions and the mechanism used for generating the feedback are the main features that distinguish various model predictive control methodologies. Cutler and Ramaker (1979, 1980) apply step response models for the prediction and incorporates feedback by update of a bias term. Richalet *et al.* (1976, 1978) apply impulse response models for the prediction and also compensates for unmeasured disturbances and plant model mismatch by update of a bias term. Feedback by update of a bias term corresponds to output disturbance models in which the disturbance is assumed to be a step. This feedback mechanism is able to achieve steady state offset free control. Greco *et al.* (1984) and Mosca (1995) apply ARX as well as ARMAX models for the prediction. Kalman filters are used to obtain feedback and steady state offset control is assured by using a velocity ARMAX formulation. Clarke *et al.* (1987a,b) as well as Bitmead *et al.* (1990) also use velocity form ARMAX models for prediction and feedback. Muske and Rawlings (1993a) apply discrete-time state space models for control and assure steady state offset free control by disturbance modelling (Muske and Badgwell, 2002; Pannocchia and Rawlings, 2003). Nørgaard *et al.* (2000) apply a neural network for predictions. Allgöwer *et al.* (1999), Allgöwer and Zheng (2000), Betts (2001) Grötschel *et al.* (2001), and Tenny (2002) discuss predictive control based on nonlinear state space models. In the following, we provide an overview of various forms of predictive control.

### 1.1.1 Linear Time Invariant Model Representations

In this subsection, linear time invariant models that are used for predictive control are informally introduced.

The discrete linear time-invariant state space model

$$x_{k+1} = Ax_k + Bu_k + w_k \quad (1.5a)$$

$$y_k = Cx_k + v_k \quad (1.5b)$$

with stochastic process noise,  $w_k$ , and measurement noise,  $v_k$  is for many situations the most convenient way to represent multiple time series  $y_k \in \mathbb{R}^p$  influenced by some signals  $u_k \in \mathbb{R}^m$  which can be manipulated.

The state,  $x_k$ , may be represented in terms of the initial state  $x_0$ , the past inputs,  $\{u_i\}_{i=0}^{k-1}$ , and the past process noise,  $\{w_i\}_{i=0}^{k-1}$

$$x_k = A^k x_0 + \sum_{i=0}^{k-1} A^{k-1-i} B u_i + \sum_{i=0}^{k-1} A^{k-1-i} w_i \quad (1.6)$$

By this expression for the state, it is obvious that the output,  $y_k$ , may be related to the initial state,  $x_0$ , the past inputs,  $\{u_i\}_{i=0}^{k-1}$ , the past process noise,

$\{w_i\}_{i=0}^{k-1}$ , and the current measurement noise,  $v_k$ , by

$$\begin{aligned} y_k &= CA^k x_0 + \sum_{i=0}^{k-1} CA^{k-1-i} Bu_i + \sum_{i=0}^{k-1} CA^{k-1-i} w_i + v_k \\ &= CA^k x_0 + \sum_{i=0}^{k-1} CA^i Bu_{k-1-i} + \sum_{i=0}^{k-1} CA^i w_{k-1-i} + v_k \\ &= CA^k x_0 + \sum_{i=0}^{k-1} CA^i Bu_{k-1-i} + n_k \end{aligned} \quad (1.7)$$

in which the measured output noise,  $n_k$ , is a combination of past process noise and current measurement noise

$$n_k = \sum_{i=0}^{k-1} CA^i w_{k-1-i} + v_k \quad (1.8)$$

The state space formulation (1.5) and associated output prediction (1.7) may be expressed as

$$y_k = CA^k x_0 + \sum_{i=1}^k H_i u_{k-i} + n_k \quad (1.9)$$

in which  $H_i$  are the impulse response coefficients defined by

$$H_i = CA^{i-1} B \quad i = 1, 2, \dots, k \quad (1.10)$$

For strictly stable systems and sufficiently large values of  $i$ , say  $i > N$ , the impulse response coefficient vanishes. Similarly, for  $k$  sufficiently large the term associated with the initial state vanishes. Therefore, the output response of linear time invariant system may be characterized by a finite number of impulse response coefficients,  $\{H_i\}_{i=1}^N$ , as in

$$y_k = \sum_{i=1}^N H_i u_{k-i} + n_k \quad \text{FIR} \quad (1.11)$$

Conceptually, in a noise free environment,  $n_k = 0$ , the impulse response coefficients,  $\{H_i\}_{i=1}^N$ , may be obtained by monitoring the outputs,  $\{y_k\}_{k=1}^N$ , after adding a unit impulse,  $u_0 = 1$ , to each input in succession.

The step response coefficients,  $S_i$ , of the linear time invariant system (1.5) are defined as

$$S_0 = 0 \quad (1.12a)$$

$$S_i = \sum_{j=1}^i CA^{j-1} B = \sum_{j=1}^i H_j \quad i = 1, 2, \dots, k \quad (1.12b)$$



Let the initial state be  $x_0 = 0$  and let the system be noise free, i.e.  $w_k = 0$  and  $v_k = 0$ . Then the step response coefficients,  $S_i$ , may be obtained by observing the output of a step response experiment for each input in succession. The impulse response coefficients,  $H_i$ , may be obtained from the step response coefficients,  $S_i$ , by

$$H_i = S_i - S_{i-1} \quad i = 1, 2, \dots, k \quad (1.13)$$

Define  $\Delta u_k = u_k - u_{k-1}$ . The output,  $y_k$ , of the linear time invariant state space model (1.5) given by (1.7) as well as by (1.9) may be expressed as

$$y_k = CA^k x_0 + S_k u_0 + \sum_{i=1}^{k-1} S_i \Delta u_{k-i} + n_k \quad (1.14)$$

using the step response coefficients,  $S_i$ . For strictly stable systems, sufficiently large  $k$ , and sufficiently long horizon,  $N$ , the step response model may be approximated by a finite set of step response coefficients,  $\{S_i\}_{i=1}^N$ , as indicated by the expression

$$y_k = S_N u_{k-N} + \sum_{i=1}^{N-1} S_i \Delta u_{k-i} + n_k \quad \text{FSR} \quad (1.15)$$

Another useful specialization of the general state space model is the state space model in innovations form. For stochastic stationary system, the state space model (1.5) may be represented as

$$x_{k+1} = Ax_k + Bu_k + Ke_k \quad (1.16a)$$

$$y_k = Cx_k + e_k \quad (1.16b)$$

in which  $e_k$  is the innovations. Kailath *et al.* (2000) provide a detailed description about the merits of the innovations form of the state space model.

The models used by the predictive controller is obtained by system identification. System identification is a very important aspect of predictive control, but is not the main topic of this thesis. Appendix D describes regression techniques that may be used to obtain impulse and step response models as well as ARX models. Appendix E describes prediction error methods aimed at obtaining models for linear model based predictive control.

### 1.1.2 Process Control Literature

Model algorithmic control (MAC) and its commercial implementation, IDCOM (identification and command) has along with DMC (dynamic matrix control) had a tremendous influence on the reputation of model predictive control. Motivated by the lack of sufficiently accurate models in the process industries, both

the IDCOM algorithm and the DMC algorithm are based on linear empirical non-parametric models; IDCOM is based a finite impulse response model while DMC is based on a finite step response model. Garcia *et al.* (1989), Qin and Badgwell (1996, 2000), Camacho and Bordons (1999), and Allgöwer *et al.* (1999) discuss the evolution of model predictive control in the process industries.

### 1.1.2.1 IDCOM Algorithms

IDCOM applies a finite impulse response (FIR) model as its predictor and a quadratic objective function (Richalet *et al.*, 1976, 1978; Rouhani and Mehra, 1982). The original IDCOM algorithm (Richalet *et al.*, 1976) solved the open-loop optimal control problem using a least-squares algorithm applied to the equality constrained problem and treated the constraints in a heuristic manner. Richalet *et al.* (1976) noted that the control problem is the dual of the model identification (parameter estimation) problem for finite impulse response models. In a modern interpretation, the IDCOM algorithm may be regarded as a moving horizon controller that at each sample instant solves the following open-loop optimal control problem

$$\min_{\{u_{k+j|k}\}_{j=0}^{N_p}} \phi = \sum_{j=0}^{N_p} \|\hat{y}_{k+j|k} - w_{k+j}\|_Q^2 + \|u_{k+j|k} - u_s\|_R^2 \quad (1.17a)$$

$$s.t. \quad \hat{y}_{k+j|k} = \sum_{i=1}^N H_i u_{k+j-i|k} + \hat{n}_{k+j|k}, \quad j = 0, 1, \dots, N_p \quad (1.17b)$$

$$u_{\min} \leq u_{k+j|k} \leq u_{\max} \quad j = 0, 1, \dots, N_p \quad (1.17c)$$

$$-\Delta_u \leq u_{k+j|k} - u_{k+j-1|k} \leq \Delta_u \quad j = 0, 1, \dots, N_p \quad (1.17d)$$

$$y_{\min} \leq \hat{y}_{k+j|k} \leq y_{\max} \quad j = 0, 1, \dots, N_p \quad (1.17e)$$

The reference trajectory,  $w_{k+j}$ , used in the objective function is computed as a first order process from the current output,  $y_k$ , to the desired set-point,  $r_{k+j} = r$

$$w_k = y_k \quad (1.18a)$$

$$w_{k+j} = \alpha w_{k+j-1} + (1 - \alpha)r_{k+j} \quad j = 1, 2, \dots, N_p \quad (1.18b)$$

$\alpha \in [0, 1[$  is a tuning parameter used to specify the desired speed of response. The closer  $\alpha$  is to 1 the slower the response to set-point deviations. The disturbances,  $\hat{n}_{k+j|k}$ , can either be estimated as constant disturbances

$$\hat{n}_{k+j|k} = \hat{n}_{k|k} = y_k - \sum_{i=1}^N H_i u_{k-i} \quad j = 0, 1, 2, \dots, N_p \quad (1.19)$$

or as smooth first order filtered disturbances (Garcia *et al.*, 1989)

$$\hat{n}_{k|k} = 0 \quad (1.20a)$$

$$\hat{n}_{k+j+1|k} = \beta \hat{n}_{k+j|k} + (1 - \beta) \left( y_k - \sum_{i=1}^N H_i u_{k-i} \right) \quad (1.20b)$$

with  $0 \leq \beta < 1$ . The penalty term on  $u_{k+j|k}$  in the MAC controller has been suggested by Garcia *et al.* (1989) as it simplifies the treatment of non-minimum phase systems.  $u_s$  may be computed as the solution of a static target calculation (c.f. Muske and Rawlings, 1993a; Muske, 1995). Alternatively, the objective function may be changed slightly

$$\phi = \sum_{j=0}^{N_p} \|\hat{y}_{k+j|k} - w_{k+j}\|_Q^2 + \|\Delta u_{k+j|k}\|_S^2 \quad (1.21)$$

in which  $\Delta u_{k+j|k} = u_{k+j|k} - u_{k+j-1|k}$  for  $j = 0, 1, \dots, N_p$ . By the objective function, (1.21), excessive control movements may be penalized without sacrificing offset free behavior and without solving a static target compensation problem.

The IDCOM algorithm was further developed by Adersa and Setpoint, Inc. Adersa marketed the new algorithm as HIECON (Hierarchical Constraint Control) and Setpoint marketed it as IDCOM-M (Grosdidier *et al.*, 1988; Qin and Badgwell, 1996). Like IDCOM, IDCOM-M is based on an impulse response model. However, IDCOM-M is able to handle integrating processes by considering variables in the  $\Delta$ -domain (i.e.  $\Delta y_k = y_k - y_{k-1}$  and  $\Delta u_k = u_k - u_{k-1}$ ). Unlike the previous industrial implementations, IDCOM-M first solve a problem with only the outputs in the objective function and subsequently if additional degrees of freedom exists an objective function with only the inputs. For cases, when the objective function in the inputs only are solved, they are sought driven as close as possible to *ideal resting values* which may come from a steady-state optimizer. The constraints in IDCOM-M are divided into soft and hard constraints. Hard constraints are ranked in order of priority. When the quadratic program constituting the moving horizon controller becomes infeasible, the lowest priority hard constraint is dropped and the calculation is repeated. Further, IDCOM-M includes a controllability supervisor used to determine online which outputs can be independently controlled. In short, IDCOM-M is regarded as third generation MPC technology which can represent and control integrating processes, allow for a control structure that changes online, distinguish between several levels of constraints (hard, ranked, soft), and provide some mechanism to recover from an infeasible solution.

The distinguishing feature characterizing the IDCOM class of moving horizon controllers is that a finite impulse response model is used to predict the future system behavior. By being based on finite impulse response models this class of moving horizon controllers are limited to stable processes or in the case of

velocity models (delta domain models) stable processes with integrators. As the FIR model may be realized as a state space model this class of controllers is really just a special instant of linear state space model predictive controllers.

### 1.1.2.2 DMC Algorithms

Model predictive control was developed by Shell Oil in the early 1970's and an initial application in 1973. These developments had a particular significant influence on the application of model predictive control technology in the process industries. Shell called this control technology dynamic matrix control (DMC). Dynamic matrix control is an unconstrained multivariable control algorithm whose details was first presented by Cutler and Ramaker (1979, 1980). DMC was based on a linear step response model for the plant, a quadratic performance objective over a finite prediction horizon, and the future plant behavior was specified by trying to follow a setpoint as closely as possible. In the original DMC algorithm the optimal inputs are computed as the solution a least-squares problem. Prett and Gillette (1979, 1980) described an application of DMC in which the algorithm was modified to handle constraints and nonlinearities. The constraint handling in Prett and Gillette (1979, 1980) was somewhat ad hoc. Morshedi *et al.* (1985) handled constraints in a systematic way by posing the control problem to be solved in the DMC algorithm as a linear program. The resulting controller was called LDMC. Garcia and Morshedi (1986) retained the quadratic objective of the DMC algorithm and handled constraints in a systematic manner by formulating the problem as a quadratic program. This controller was called QDMC. In addition Garcia and Morshedi (1986) discussed the role of output constraints. They argued, that for some systems the hard output constraints are required to be satisfied only at portions of the horizon; called the constraint window. The constraint window typically starts at some point in the future and continues on to steady state. The soft constraint concept was also applied by Garcia and Morshedi (1986) and they described an approximate implementation.

The key distinguishing feature of the DMC class of algorithms is that their predictions are based on step response models. The quadratic program constituting the open-loop optimal control problem in the DMC implementa-

tion of moving horizon control is

$$\min_{\{\Delta u_{k+j|k}\}_{j=0}^{M-1}} \phi = \sum_{j=0}^{N_p} \|\hat{y}_{k+j|k} - r_{k+j}\|_{Q_j}^2 + \sum_{j=0}^{M-1} \|\Delta u_{k+j|k}\|_{S_j}^2 \quad (1.22a)$$

$$s.t. \quad \hat{y}_{k+j|k} = \sum_{i=1}^N S_i \Delta u_{k+j-i|k} + S_N u_{k+j-N|k} + \hat{n}_{k+j|k} \quad (1.22b)$$

$$u_{\min} \leq u_{k+j|k} \leq u_{\max} \quad j = 0, 1, \dots, N_p \quad (1.22c)$$

$$-\Delta u \leq \Delta u_{k+j|k} \leq \Delta u \quad j = 0, 1, \dots, M-1 \quad (1.22d)$$

$$\Delta u_{k+j|k} = 0 \quad j = M, M+1, \dots, N_p \quad (1.22e)$$

$$y_{\min} \leq \hat{y}_{k+j|k} \leq y_{\max} \quad j = 0, 1, \dots, N_p \quad (1.22f)$$

in which  $\Delta u_{k+i|k} = u_{k+i|k} - u_{k+i-1|k}$  and

$$u_{k+j|k} = u_{k-1|k} + \sum_{i=0}^j \Delta u_{k+i|k} \quad (1.23)$$

The disturbance is estimated as a constant output disturbance

$$\hat{n}_{k+j|k} = \hat{n}_{k|k} = y_k - \sum_{i=1}^N S_i \Delta u_{k-i|k} + S_N u_{k-N|k} \quad (1.24)$$

Dynamic matrix control distinguishes between a control horizon,  $M$ , and a prediction horizon,  $N_p > M$ . The input movements,  $\Delta u_k$ , can change within the control horizon, i.e.  $k = 0, 1, \dots, M-1$ , but are fixed beyond the control horizon, i.e.  $\Delta u_k = 0$  for  $k = M, M+1, \dots, N_p$ . The prediction horizon is longer than the control horizon and chosen such that the open-loop response computed as the solution of (1.22) for long control horizons resembles the closed-loop response.

Soft constraints are incorporated in the QDMC formulation (1.22) by introduction of slack variables,  $p_j$  and  $q_j$ , and replacement of the hard output constraints (1.22f) by the constraints

$$\hat{y}_{k+j|k} + p_j \geq y_{\min} \quad j = 0, 1, \dots, N_p \quad (1.25a)$$

$$\hat{y}_{k+j|k} - q_j \leq y_{\max} \quad j = 0, 1, \dots, N_p \quad (1.25b)$$

$$p_j \geq 0 \quad j = 0, 1, \dots, N_p \quad (1.25c)$$

$$q_j \geq 0 \quad j = 0, 1, \dots, N_p \quad (1.25d)$$

as well as substitution of the objective function (1.22a) by

$$\phi = \sum_{j=0}^{N_p} \|\hat{y}_{k+j|k} - r_{k+j}\|_{Q_j}^2 + \sum_{j=0}^{M-1} \|\Delta u_{k+j|k}\|_{S_j}^2 + \sum_{j=0}^{N_p} \|p_j\|_{W_j}^2 + \|q_j\|_{W_j}^2 \quad (1.26)$$

As the prediction in the DMC algorithms are based on finite step response models, they are limited to stable processes in case when the outputs of the model are the variables  $y_k$ . If the model is identified in the delta-domain, i.e. the outputs are  $\Delta y_k = y_k - y_{k-1}$  the DMC algorithms may be used to control processes with integrators. The DMC algorithms are special cases of state space based model predictive controllers as the step response model may be converted to a state space model.

### 1.1.3 Generalized Predictive Control

Clarke *et al.* (1987a,b) presented an alternative predictive control methodology that they called generalized predictive control (GPC). The generalized predictive control methodology had its origins in self-tuning control community as and was supposed to replace generalized minimum variance control and pole placement. GPC has found a market niche in providing a justifiable, sophisticated yet flexible and comprehensible adaptive controller design platform.

The generalized predictive controller is based on an ARMAX model for prediction of the future evolution of the system. The ARMAX model used is also referred to as a controlled auto regressive and integrated moving average model (CARIMA) or auto-regressive integrated moving-average exogeneous input (ARIMAX) model. This model has the structure

$$A(q^{-1})y_k = B(q^{-1})u_k + C(q^{-1})e_k \quad (1.27)$$

in which  $A$  and  $B$  are polynomials in the backward shift operator,  $q^{-1}$ .  $A$  is a monic polynomial and the model is integrated as  $C$  has the structure

$$C(q^{-1}) = \frac{T(q^{-1})}{1 - q^{-1}} \quad (1.28)$$

in which  $T$  is a monic polynomial in the backward shift operator  $q^{-1}$ . The operator,  $\Delta = 1 - q^{-1}$ , is used to achieve offset free regulation when step output disturbances occur. This model is equivalent to the following model in the  $\Delta$ -domain

$$A(q^{-1})\Delta y_k = B(q^{-1})\Delta u_k + T(q^{-1})e_k \quad (1.29)$$

or as

$$\tilde{A}(q^{-1})y_k = \tilde{B}(q^{-1})u_k + \tilde{C}(q^{-1})e_k \quad (1.30)$$

in which

$$\tilde{A}(q^{-1}) = (1 - q^{-1})A(q^{-1}) \quad (1.31a)$$

$$\tilde{B}(q^{-1}) = (1 - q^{-1})B(q^{-1}) \quad (1.31b)$$

$$\tilde{C}(q^{-1}) = T(q^{-1}) \quad (1.31c)$$

For this class of input-output models, the GPC and systems identification literature (c.f. Clarke, 1994; Ljung, 1999), there has been a tradition of formulating

the predictors in terms of backward shift operators. Another approach is to convert the input-output model (1.30) to state space form and derive the predictors from the state space model. Consider one input-output pair of (1.30)

$$\tilde{A}(q^{-1}) = 1 + \tilde{a}_1 q^{-1} + \tilde{a}_2 q^{-2} + \dots + \tilde{a}_n q^{-n} \quad (1.32a)$$

$$\tilde{B}(q^{-1}) = \tilde{b}_0 + \tilde{b}_1 q^{-1} + \tilde{b}_2 q^{-2} + \dots + \tilde{b}_n q^{-n} \quad (1.32b)$$

$$\tilde{C}(q^{-1}) = 1 + \tilde{c}_1 q^{-2} + \tilde{c}_2 q^{-2} + \dots + \tilde{c}_n q^{-n} \quad (1.32c)$$

for which the state space model

$$x_{k+1} = Ax_k + Bu_k + Ke_k \quad (1.33a)$$

$$y_k = Cx_k + Du_k + e_k \quad (1.33b)$$

in observer canonical realization is (c.f. Poulsen, 1995a; Goodwin and Sin, 1984)

$$A = \begin{bmatrix} -\tilde{a}_1 & 1 & \dots & 0 \\ \vdots & & \ddots & \\ -\tilde{a}_{n-1} & 0 & \dots & 1 \\ -\tilde{a}_n & 0 & \dots & 0 \end{bmatrix} \quad B = \begin{bmatrix} \tilde{b}_1 - \tilde{a}_1 \tilde{b}_0 \\ \tilde{b}_2 - \tilde{a}_2 \tilde{b}_0 \\ \vdots \\ \tilde{b}_n - \tilde{a}_n \tilde{b}_0 \end{bmatrix} \quad K = \begin{bmatrix} \tilde{c}_1 - \tilde{a}_1 \\ \tilde{c}_2 - \tilde{a}_2 \\ \vdots \\ \tilde{c}_n - \tilde{a}_n \end{bmatrix} \quad (1.34a)$$

$$C = [1 \quad 0 \quad \dots \quad 0] \quad D = \tilde{b}_0 \quad (1.34b)$$

The observer canonical realization (1.34) is simplified and identical to (1.5) in some sense when  $\tilde{b}_0 = 0$  as is often the case in discrete time systems. Furthermore, the expression for the  $B$ -matrix simplifies considerably when  $\tilde{b}_0 = 0$ . The realization, (1.33) and (1.34), is extended to the MISO case by having a column in the  $B$  and  $D$  matrix for each input constructed in the way indicated above. The MIMO model is constructed by stacking the MISO realizations for each output. These models are not necessarily minimal. A minimal realization may be obtained by construction of the associated Markov parameters and realization of the state space model from these parameters. A realization of (1.30) in state space innovations form

$$x_{k+1} = Ax_k + Bu_k + Ke_k \quad (1.35a)$$

$$y_k = Cx_k + Du_k + e_k \quad (1.35b)$$

in which  $e_k$  is a white noise sequence with covariance  $E\{e_k e_k'\} = R$  may be used to form the Kalman filter estimate of the current state and the Kalman prediction of the future evolution of the system. This way of approaching GPC is untraditional as the predictors are typically realized through Diophantine equations and input-output operators rather than through state space prediction employing Kalman's methods. The Kalman filter for the state space model in innovations form and a certain initial state (i.e. we know the past measured outputs and inputs exactly) is particularly simple. The filtered state estimate,

$\hat{x}_{k|k}$ , is  $\hat{x}_{k|k} = \hat{x}_{k|k-1}$  and the innovations  $\varepsilon_k$  are computed by

$$\hat{y}_{k|k-1} = C\hat{x}_{k|k-1} + D\hat{u}_{k|k-1} \quad (1.36a)$$

$$\varepsilon_k = y_k - \hat{y}_{k|k-1} \quad (1.36b)$$

The innovations  $\varepsilon_k$ ,  $\hat{x}_{k|k-1}$ , and  $\{\hat{u}_{k+j|k}\}$  are used to compute the optimal state predictions  $\{\hat{x}_{k+j|k}\}$

$$\hat{x}_{k+1|k} = A\hat{x}_{k|k-1} + B\hat{u}_{k|k} + K\varepsilon_k \quad (1.37a)$$

$$\hat{x}_{k+j+1|k} = A\hat{x}_{k+j|k} + B\hat{u}_{k+j|k} \quad j = 1, 2, \dots \quad (1.37b)$$

The optimal predictions  $\{\hat{y}_{k+j|k}\}$  of the outputs  $\{y_{k+j}\}$  are computed by

$$\hat{y}_{k+j|k} = C\hat{x}_{k+j|k} + D\hat{u}_{k+j|k} \quad j = 1, 2, \dots \quad (1.38)$$

Consequently, the predictions  $\{\hat{y}_{k+j|k}\}$  may be computed from the state space system constituted by (1.37) and (1.38). The optimal output predictions,  $\{\hat{y}_{k+j|k}\}_{j=N_1}^{N_2}$ , of this system may be expressed explicitly as function of the predicted state,  $\hat{x}_{k|k-1}$ , the innovation  $\varepsilon_k$ , and the predicted outputs  $\{\hat{u}_{k+j|k}\}$  as

$$\hat{y}_{k+j|k} = CA^j\hat{x}_{k|k-1} + CA^{j-1}K\varepsilon_k + \sum_{i=0}^{j-1} CA^{j-1-i}B\hat{u}_{k+i|k} + D\hat{u}_{k+j|k} \quad (1.39)$$

for  $j = N_1, N_1 + 1, \dots, N_2$ .

The objective minimized by the generalized predictive controller consist of term accounting for setpoint deviations and a term penalizing excessive control movements

$$\phi = \frac{1}{2} \sum_{j=N_1}^{N_2} (\hat{y}_{k+j|k} - r_{k+j})' Q_j (\hat{y}_{k+j|k} - r_{k+j}) + \frac{1}{2} \sum_{j=0}^{N_u} \Delta \hat{u}'_{k+j|k} S_j \Delta \hat{u}_{k+j|k} \quad (1.40)$$

in which  $\Delta \hat{u}_{k+j|k} = \hat{u}_{k+j|k} - \hat{u}_{k+j-1|k}$ . The horizons,  $N_1$ ,  $N_2$ , and  $N_u$  are used as tuning parameters. However, the effect of these parameters are non-intuitive. Current knowledge suggests, that  $N_1 = 0$  and  $N_2 = N_u = \infty$  if nominal stability is to be guaranteed. A practical and common approach is to select  $N_2$  and  $N_u$  as large integer values and such that  $N_2 > N_u$ .

The mathematical program solved online at each sampling instant in the generalized predictive control methodology is

$$\min_{\{\hat{u}_{k+j|k}\}} \phi \quad (1.41a)$$

$$s.t. \quad \hat{y}_{k+j|k} \text{ computed by (1.39)} \quad j = N_1, N_1 + 1, \dots, N_2 \quad (1.41b)$$

$$u_{\min} \leq \hat{u}_{k+j|k} \leq u_{\max} \quad j = 0, 1, \dots, N_u - 1 \quad (1.41c)$$

$$-\Delta_u \leq \Delta \hat{u}_{k+j|k} \leq \Delta_u \quad j = 0, 1, \dots, N_u - 1 \quad (1.41d)$$

$$\Delta \hat{u}_{k+j|k} = 0 \quad j = N_u + 1, \dots, N_2 \quad (1.41e)$$



The main point to be emphasized regarding the mathematical program (1.41) is that although the predictions are stated explicitly and arise from an input-output model (1.27), the predictions may be stated in terms of a state space model (1.37)-(1.38). Consequently, for long prediction horizons, the solution of (1.41) may be conducted efficiently using algorithms utilizing the state space structure. The conventional approach in GPC is to construct a dense quadratic program and compute the predicted optimal controls as the solution of that program. Feedback is achieved using the Kalman filter. For state space systems in innovations form this is conducted by updating the memory,  $\hat{x}_{k+1|k}$  and  $\hat{u}_{k+1|k}$ , of the controller using (1.37a) and  $\hat{u}_{k+1|k}$  computed as part of the solution to (1.41). At the next sample time, i.e.  $k+1$ , the measurement  $y_{k+1}$  and the memory are used to compute the innovation  $\varepsilon_{k+1}$ . With the innovation  $\varepsilon_{k+1}$  as well as the memory,  $\hat{x}_{k+1|k}$  and  $\hat{u}_{k+1|k}$ , the new control profile  $\{\hat{u}_{k+1+j|k+1}\}$  may be computed and the first control in this sequence implemented on the process, i.e.  $u_{k+1} = \hat{u}_{k+1|k+1}$ . It should also be stressed that a central premise of this controller is that it is based on a process model augmented with a constant output disturbance model.

The generalized predictive controller (1.41) may be extended by output constraints

$$y_{\min} \leq \hat{y}_{k+j|k} \leq y_{\max} \quad (1.42)$$

These constraints may render the mathematical program constituting the generalized predictive controller, i.e. (1.41)-(1.42), infeasible. To overcome such problems, the controller must have some mechanism of relaxing the mathematical program (1.41)-(1.42). This may be achieved by soft constraints.

Generalized predictive control has enjoyed popularity in the adaptive control community as it is based on input-output models of the ARMAX type which are parameterizations with the fewest possible parameters. Generalized predictive control has been discussed extensively by Bitmead *et al.* (1990), Soeterboek (1992), Mosca (1995), Martin-Sanchez and Rodellar (1996), and Camacho and Bordons (1999).

Hallager *et al.* (1984), Hallager (1984), and Brabrand (1991) developed a multivariable self tuning controller (MIMOSC) which is similar to the generalized predictive control methodology. MIMOSC models the system using multivariable ARMAX models estimated recursively and designs the controller as an LQ controller. Input constraints are handled by clipping. MIMOSC has been applied to control of a fixed bed reactor (Hallager *et al.*, 1984; Hallager, 1984; Brabrand, 1991) as well as in the control of a heat-integrated distillation column (Andersen, 1989; Nielsen, 1990; Fikar and Jørgensen, 1994; Koggersbøl, 1995; Andersen, 2002).

### 1.1.4 Discrete-Time State Space Models and Kalman

The IDCOM, the DMC and the GPC class of predictive controllers all have their roots and formulation in the associated model identification algorithm such that the connection and fitting to the data of the specific process is straightforward. IDCOM is based on an impulse response representation of the plant, the prediction in DMC is based on a step response model, and GPC applies an integrated ARMAX model to predict the evolution of the plant. The possibility of matching the model based on data to the specific process considered has been a significant factor for the popularity of the above predictive control methods. The impulse response model, the step response model, and the integrated ARMAX model may all be represented as state space models (Li *et al.*, 1989; Lee *et al.*, 1992, 1994). This representation may be stated explicitly through the parameters of the input-output models or may be constructed through the Hankel matrix of the system.

At the conception of the above predictive control techniques, they were considered different from the linear-quadratic regulation methodology developed by Kalman (Kalman, 1960a,b; Kalman and Bertram, 1960). In the process industries linear-quadratic regulation was considered impractical as it was not obvious how to obtain the models used by the Kalman-filter and in the linear-quadratic regulator. Though theoretically possible, development of nonlinear first principles process models, linearization and discretization was neither an efficient nor a practical method meeting the accuracy requirements in controlling industrial processes. The strength of the state space method advocated by Kalman is its versatility, its direct and straightforward implementation in software, and its value as a framework for analyzing properties such as stability. For these reasons and the fact that input-output models may be represented as state space models, contemporary practice is to discuss model predictive control in terms of state space models. The state space models applied may be obtained from non-linear process models through linearization and discretization, as realizations of input-output models, or identified directly as in the subspace methods (van Overschee and de Moor, 1996; Larimore, 1983).

Shell (France) has developed third generation MPC technology called SMOC which is similar to modern model predictive control implementations (Marquis and Broustail, 1988). SMOC is based on state space models which can represent both stable and unstable models. Feedback is achieved by full state estimation from the outputs; constant output disturbances is simply a special case in this methodology. Further, SMOC distinguishes between controlled variables that are in the control objective function and feedback variables that are used for state estimation. This distinction of variables may be used in the implementation of inferential control. As an example of inferential control, consider distillation composition control which may be realized by having the product compositions in the objective function even though feedback is based on temperature and pressure measurements. The compositions are the controlled

variables while the temperature and pressures are the feedback variables. However, during model identification samples of the compositions must be analyzed to establish the correlation between the measured and the controlled variables. During normal operation with this controller, infrequent compositions samples may also be helpful in regular calibration of the composition model. SMOC handles input and output constraints through a QP-formulation of the controller.

Based on a state-space representation, Bortolotto (1985) and Bortolotto and Jørgensen (1986) showed that the predictive controller may be solved as a dense quadratic program. Li *et al.* (1989) as well as Lee *et al.* (1992, 1994) demonstrated how step-response models may be represented as state space models. Ricker (1990) developed a finite-horizon predictive controller and applied dense quadratic programming algorithms for its solution (Ricker, 1985). Even though Ricker (1990) is aware of the possibility of feedback by Kalman filtering as described in Li *et al.* (1989), he rejects this solution and essentially constructs feedback based on a constant output disturbance model. The motivation for doing so is that sufficiently accurate covariance matrices needed for the Kalman filter are seldom available according to Ricker (1990). Based on a state-space model representation, Muske and Rawlings (1993a) provided an extraordinary nice discussion of model predictive control and showed how nominally stabilizing predictive controller may be constructed. Muske and Rawlings (1993a) had feedback by employment of a static Kalman filter. Scokaert and Rawlings (1998) extended the results of Muske and Rawlings (1993a) and provided a better and more efficient parameterization of the predictive controller.

The model used for prediction in linear state space based predictive controllers is a linear discrete time state space model which may be deterministic or stochastic. However, the deterministic model is typically augmented by stochastic terms to facilitate feedback by application of a Kalman filter. For these reasons, we consider the predictive control based on discrete-time state space models for the stochastic model

$$x_{k+1} = Ax_k + Bu_k + w_k \quad (1.43a)$$

$$y_k = Cx_k + v_k \quad (1.43b)$$

in which

$$\begin{bmatrix} w_k \\ v_k \end{bmatrix} \in N_{iid} \left( \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} Q & S \\ S' & R \end{bmatrix} \right) \quad (1.44)$$

Let  $x_0 \in N(\hat{x}_{0|-1}, P_{0|-1})$  be independent of  $\{w_k\}$  and  $\{v_k\}$ . The theory of linear estimation (Kailath *et al.*, 2000) may be used to construct the optimal predictors for this model based on the past measurements  $\{y_j\}_{j=0}^k$  and past control actions  $\{u_j\}_{j=0}^{k-1}$ . The optimal predictors are themselves generated by a deterministic linear state space model.

As a new measurement,  $y_k$ , at discrete time  $k$  is fed into the estimator, the optimal estimate,  $\hat{x}_{k|k}$ , of the current state is obtained by first computing the

innovation,  $e_k$ , according to

$$\hat{y}_{k|k-1} = C\hat{x}_{k|k-1} \quad (1.45a)$$

$$e_k = y_k - \hat{y}_{k|k-1} \quad (1.45b)$$

and subsequently computing the Kalman filter gain,  $K_{f,k}$  according to

$$R_{e,k} = CP_{k|k-1}C' + R \quad (1.46a)$$

$$K_{f,k} = P_{k|k-1}C'R_{e,k}^{-1} \quad (1.46b)$$

With the innovation,  $e_k$ , and the Kalman filter gain,  $K_{f,k}$ , available, the current state estimate  $\hat{x}_{k|k}$  and the associated covariance,  $P_{k|k}$ , may be obtained by

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_{f,k}e_k \quad (1.47a)$$

$$P_{k|k} = P_{k|k-1} - K_{f,k}R_{e,k}K_{f,k}' \quad (1.47b)$$

Similarly, the current estimate,  $\hat{w}_{k|k}$ , of the process noise and its associated covariance may be computed as

$$\hat{w}_{k|k} = SR_{e,k}^{-1}e_k \quad (1.48a)$$

$$Q_{k|k} = Q - SR_{e,k}^{-1}S' \quad (1.48b)$$

In cases when the process and measurement noise, i.e.  $w_k$  and  $v_k$ , are uncorrelated ( $S = 0$ ), the process noise estimate is  $\hat{w}_{k|k} = 0$  and  $Q_{k|k} = Q$ .

The estimates,  $\hat{x}_{k|k}$  and  $\hat{w}_{k|k}$ , are used to form the optimal predictions of the future evolution of the system. The corresponding covariances are used to form the covariances associated with the predictors. As the system is linear and normally distributed these properties provide a complete characterization of the stochastic system. The optimal one-step predictor of the states and its covariance are computed by

$$\hat{x}_{k+1|k} = A\hat{x}_{k|k} + B\hat{u}_{k|k} + \hat{w}_{k|k} \quad (1.49a)$$

$$P_{k+1|k} = AP_{k|k}A' + Q_{k|k} - AK_{f,k}S' - SK_{f,k}'A' \quad (1.49b)$$

These expressions simplify in the case when the process and measurement noise are uncorrelated as  $S = 0$ ,  $Q_{k|k} = Q$ , and  $\hat{w}_{k|k} = 0$ . The covariance matrix  $P_{k+1|k}$  converges to the matrix  $P$  under the assumption that  $(A, C)$  is detectable and  $(A - SR^{-1}C, Q - SR^{-1}S')$  is stabilizable. The matrix  $P$  may be computed as the solution of the Riccati equation

$$P = Q + APA' - (APC' + S)(CPC' + R)^{-1}(APC' + S)' \quad (1.50)$$

which may subsequently be used to compute the following matrices

$$R_e = CPC' + R \quad (1.51a)$$

$$K_f = PC'R_e^{-1} = PC'(CPC' + R)^{-1} \quad (1.51b)$$

$$K_w = SR_e^{-1} = S(CPC' + R)^{-1} \quad (1.51c)$$

$R_e$  and  $K_f$  are the converged equivalents of  $R_{e,k}$  and  $K_{f,k}$ , respectively. Under the assumption that the Kalman filter has converged, the computation of the optimal estimates,  $\hat{x}_{k|k}$  and  $\hat{w}_{k|k}$ , may be simplified to the expressions

$$e_k = y_k - C\hat{x}_{k|k-1} \quad (1.52a)$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_f e_k \quad (1.52b)$$

$$\hat{w}_{k|k} = K_w e_k \quad (1.52c)$$

Whether  $\hat{x}_{k|k}$  and  $\hat{w}_{k|k}$  are computed by the dynamic or the static Kalman filter, they can be used to form the one-step prediction in (1.49a). The one-step predictor and its associated covariance,  $P_{k+1|k}$  (or  $P$  in the static case), are the starting point for the  $j$ -step predictor of the states and its associated covariance

$$\hat{x}_{k+j+1|k} = A\hat{x}_{k+j|k} + B\hat{u}_{k+j|k} \quad j = 1, 2, \dots \quad (1.53a)$$

$$P_{k+j+1|k} = AP_{k+j|k}A' + Q \quad j = 1, 2, \dots \quad (1.53b)$$

The predictions of the future measurements and their covariances are computed by

$$\hat{y}_{k+j|k} = C\hat{x}_{k+j|k} \quad j = 1, 2, \dots \quad (1.54a)$$

$$\Lambda_{k+j|k} = CP_{k+j|k}C' + R \quad j = 1, 2, \dots \quad (1.54b)$$

In many practical situations, such as inferential control, a distinction is made between the measurement variables,  $y_k$ , used for feedback and the variables, say  $z_k$ , controlled. Let the controlled variables be related to the states by

$$z_k = Ex_k \quad (1.55)$$

Then the optimal predictions,  $\hat{z}_{k+j|k}$ , and the associated covariances,  $W_{k+j|k}$ , of the controlled variables are given by the expressions

$$\hat{z}_{k+j|k} = E\hat{x}_{k+j|k} \quad j = 0, 1, 2, \dots \quad (1.56a)$$

$$W_{k+j|k} = EP_{k+j|k}E' \quad j = 0, 1, 2, \dots \quad (1.56b)$$

(1.49a), (1.53a), and (1.56a) are used to compute the future predictions. When  $S = 0$ ,  $\hat{w}_{k|k} = 0$  and these predictors may be thought of as generated by a standard deterministic state space model

$$x_{k+1} = Ax_k + Bu_k \quad (1.57a)$$

$$z_k = Ex_k \quad (1.57b)$$

Influenced by Muske and Rawlings (1993a), the state space model (1.57) has been used extensively to discuss model predictive control and develop nominally stabilizing predictive controllers for systems modeled by linear state space models. However, the predictor system (1.57) is just a special case of the general

optimal predictor, i.e. (1.49a), (1.53a), and (1.56a), of the linear discrete-time stochastic system (1.43,1.55). In fact, as the general optimal predictor turns out to be an affine system; this general case is therefore in itself a motivation for introducing the affine state space models and develop optimal control algorithms for this class of systems. This topic is addressed in this dissertation by constructing algorithms for (1.3) and related problems.

The quality of a given predicted trajectory may be evaluated through the following objective function

$$\phi = \frac{1}{2} \sum_{j=0}^{N_p} (\hat{z}_{k+j|k} - r_{k+j})' \Xi_j (\hat{z}_{k+j|k} - r_{k+j}) + \frac{1}{2} \sum_{j=0}^{N-1} \Delta \hat{u}'_{k+j|k} \Theta_j \Delta \hat{u}_{k+j|k} \quad (1.58)$$

in which the first part penalizes predicted deviations from the reference trajectory  $\{r_{k+j}\}$  and the second part penalizes excessive control movements. The computation of this objective function requires computation of the predicted outputs,  $\hat{z}_{k+j|k}$ , which are evaluated through (1.49a), (1.53a), and (1.56a). The weights,  $\Xi_j$ , in the first part of the objective function are often a constant matrix selected through manual tuning. However, it might also be selected as  $\Xi_j = W_{k+j|k}^{-1}$  such that this part of the objective has minimum predicted variance. This selection is in accordance with the theory of stochastic least squares optimization (Kailath *et al.*, 2000). Alternatively, in light of the moving horizon implementation these weights could be selected as  $\Xi_j = W_{k+1|k}^{-1}$ . The selection of the horizons  $N_p$  and  $N$  has been subject of extensive research (Muske and Rawlings, 1993a; Scokaert and Rawlings, 1998; Mayne *et al.*, 2000).

As other controllers solved by online optimization and implemented in a moving horizon manner, predictive controllers in state space form may be equipped with hard constraints on the actuator values as well as hard constraints on the actuator rate of movements

$$u_{\min} \leq \hat{u}_{k+j|k} \leq u_{\max} \quad (1.59a)$$

$$-\Delta_u \leq \Delta \hat{u}_{k+j|k} \leq \Delta_u \quad (1.59b)$$

The predicted outputs may also be constrained

$$z_{\min} \leq \hat{z}_{k+j|k} \leq z_{\max} \quad (1.60)$$

This constraint may, however, render the predictive controller mathematical program infeasible. Therefore, output constraints must be associated with some relaxation mechanism if they are impossible to satisfy (Scokaert and Rawlings, 1999).

The control calculation in the predictive controller based on a discrete time

state space models is achieved by solution of the quadratic program

$$\min_{\{\hat{u}_{k+j|k}\}_{j=0}^{N_p}} \phi \quad (1.61a)$$

$$s.t. \quad \hat{x}_{k+1|k} = A\hat{x}_{k|k} + B\hat{u}_{k|k} + \hat{w}_{k|k} \quad (1.61b)$$

$$\hat{x}_{k+j+1|k} = A\hat{x}_{k+j|k} + B\hat{u}_{k+j|k} \quad j = 1, 2, \dots, N_p - 1 \quad (1.61c)$$

$$\hat{z}_{k+j|k} = E\hat{x}_{k+j|k} \quad j = 0, 1, \dots, N_p \quad (1.61d)$$

$$u_{\min} \leq \hat{u}_{k+j|k} \leq u_{\max} \quad j = 0, 1, \dots, N - 1 \quad (1.61e)$$

$$-\Delta_u \leq \Delta\hat{u}_{k+j|k} \leq \Delta_u \quad j = 0, 1, \dots, N - 1 \quad (1.61f)$$

$$\Delta\hat{u}_{k+j|k} = 0 \quad j = N, N + 1, \dots, N_p \quad (1.61g)$$

$$z_{\min} \leq \hat{z}_{k+j|k} \leq z_{\max} \quad j = 0, 1, \dots, N_p \quad (1.61h)$$

(1.61) is a finite-horizon predictive controller as  $N_p > N$  is finite. The parameterization applied, i.e. (1.61g), allows the actuator variables to vary during the first  $N$  samples after which it is fixed for the remaining part of the prediction horizon. The merit of this parameterization is that it is particularly simple to implement. Muske and Rawlings (1993a) have extended this parameterization to an infinite prediction horizon,  $N_p = \infty$ , and shown how to compute the solution of the associated infinite-dimensional quadratic program.

### Deterministic Continuous-Time State Space Model

The deterministic model linear discrete-time state space model may be obtained from a nonlinear model based on physical principles. Such a model may be stated in terms of ordinary differential equations

$$\dot{x}(t) = f(x(t), u(t)) \quad (1.62a)$$

$$y(t) = g(x(t)) \quad (1.62b)$$

for which the corresponding linearized dynamics around the equilibrium point  $(x_s, u_s, y_s)$  is

$$\delta\dot{x}(t) = F\delta x(t) + G\delta u(t) \quad (1.63a)$$

$$\delta y(t) = C\delta x(t) \quad (1.63b)$$

with  $\delta x(t) = x(t) - x_s$ ,  $\delta u(t) = u(t) - u_s$ , and  $\delta y(t) = y(t) - y_s$ . The linear state space matrices  $F$ ,  $G$ , and  $C$  are

$$F = \nabla_x f(x_s, u_s)' \quad (1.64a)$$

$$G = \nabla_u f(x_s, u_s)' \quad (1.64b)$$

$$C = \nabla_x g(x_s)' \quad (1.64c)$$

The discretized linearized system (in deviation variables) is

$$x_{k+1} = Ax_k + Bu_k \quad (1.65a)$$

$$y_k = Cx_k \quad (1.65b)$$

and the state space matrices  $A$  and  $B$  are given by

$$A = A(T_s) = \exp(FT_s) \quad (1.66a)$$

$$B = B(T_s) = \int_0^{T_s} \exp(Fs) G ds \quad (1.66b)$$

and may be computed using the matrix exponential function

$$\exp\left(\begin{bmatrix} F & G \\ 0 & 0 \end{bmatrix} T_s\right) = \begin{bmatrix} A(T_s) & B(T_s) \\ 0 & I \end{bmatrix} \quad (1.67)$$

### Stochastic Continuous-Time State Space Model

The stochastic discrete-time linear state space model may be derived from the linear stochastic differential equation

$$dx(t) = (Fx(t) + Gu(t)) dt + Hd\beta(t) \quad (1.68a)$$

$$y(t_k) = Cx(t_k) + v(t_k) \quad (1.68b)$$

in which  $\beta(t)$  is a standard Wiener process (Jazwinski, 1970; Åström, 1970b) and  $v(t_k) \in N_{iid}(0, R)$  is independent identically normally distributed measurement noise. The corresponding discrete-time state space model describing evolution of the mean value and used in the implementation of state space based predictive controllers may be denoted

$$x_{k+1} = Ax_k + Bu_k + w_k \quad (1.69a)$$

$$y_k = Cx_k + v_k \quad (1.69b)$$

in which the measurement noise  $v_k \in N_{iid}(0, R)$  and process noise  $w_k \in N_{iid}(0, Q)$  are both normally distributed. The matrices  $A$ ,  $B$ , and  $Q$  may be expressed as

$$A = A(T_s) = \exp(FT_s) \quad (1.70a)$$

$$B = B(T_s) = \int_0^{T_s} \exp(Fs) G ds \quad (1.70b)$$

$$Q = Q(T_s) = \int_0^{T_s} \exp(Fs) H H' \exp(F's) ds \quad (1.70c)$$

and computed efficiently using the matrix exponential

$$\exp\left(\begin{bmatrix} -F & HH' & 0 \\ 0 & F' & I \\ 0 & 0 & 0 \end{bmatrix} t\right) = \begin{bmatrix} F_1(t) & G_1(t) & H_1(t) \\ 0 & F_2(t) & G_2(t) \\ 0 & 0 & F_3(t) \end{bmatrix} \quad (1.71)$$

and the relations (Moler and Van Loan, 1978; Van Loan, 1978; Sidje, 1998; Kristensen *et al.*, 2002)

$$A = A(T_s) = F_2(T_s)' \quad (1.72a)$$

$$B = B(T_s) = G_2(T_s)' G \quad (1.72b)$$

$$Q = Q(T_s) = F_2(T_s)' G_1(T_s) \quad (1.72c)$$



### Disturbance Modeling and Offset-Free Control

Closed-loop performance of model predictive control is directly related to model accuracy. Unmeasured disturbances and modeling error can lead to steady-state offset unless the predictive control algorithm is designed for these types of plant-model mismatch. Elimination of steady state offset is achieved in two ways. The first method is inspired by PI-control and involves modifying the controller objective to include an integral of the tracking error in the objective function (Kwakernaak and Sivan, 1972; Franklin *et al.*, 1998). This method of achieving offset free control leaves the estimator unaffected while the controller is modified to include an integral term. This method implicitly assumes that the measured and tracked signals are identical. The second method to eliminate offset in a system controlled by a predictive controller involves augmenting the process model by a step disturbance model (Kwakernaak and Sivan, 1972; Joseph and Brosilow, 1978; Morari and Stephanopoulos, 1980; Franklin *et al.*, 1998; Muske and Badgwell, 2002; Pannocchia and Rawlings, 2003). In this approach, the estimator attempts to estimate the system states as well as the disturbances. Knowledge of the disturbances are applied in the controller to force a zero steady state tracking error whenever possible.

Consider the discrete-time stochastic linear model

$$x_{k+1} = Ax_k + Bu_k + w_k \quad (1.73a)$$

$$y_k = Cx_k + v_k \quad (1.73b)$$

for which the optimal filter is given by (1.47a) and (1.48a). The optimal state predictor is given by (1.49a) and (1.53a). Define the tracking error as  $e_k = y_k - r_k$  and the sum of the tracking error,  $I_k$ , as

$$I_k = \sum_{j=0}^{k-1} e_j \quad (1.74)$$

which may be expressed recursively as

$$I_{k+1} = I_k + e_k \quad I_0 = 0 \quad (1.75)$$

The optimal prediction of the integral of the tracking error is

$$\hat{I}_{k|k} = I_k \quad (1.76a)$$

$$\hat{I}_{k+1|k} = I_k + e_k \quad (1.76b)$$

$$\hat{I}_{k+j+1|k} = I_{k+j|k} + (C\hat{x}_{k+j|k} - \hat{r}_{k+j|k}) \quad j \geq 1 \quad (1.76c)$$

and may be used in evaluating the objective function of the predictive controller augmented by an integral term

$$\phi = \frac{1}{2} \sum_{j=0}^{N_p} \hat{e}'_{k+j|k} Q \hat{e}_{k+j|k} + \frac{1}{2} \sum_{j=0}^{N_p} \hat{I}'_{k+j|k} Q_I \hat{I}_{k+j|k} + \frac{1}{2} \sum_{j=0}^{N-1} \Delta \hat{u}'_{k+j|k} S \Delta \hat{u}_{k+j|k} \quad (1.77)$$

When this objective is applied in a predictive controller, the first term gives rise to the proportional part of the controller, while the second term in the objective function gives rise to the integral part of the controller. The third term in this objective function is used to make the controller well-conditioned. The predictor applied in this controller has the structure (for  $j \geq 1$ )

$$\begin{bmatrix} \hat{x}_{k+j+1|k} \\ \hat{I}_{k+j+1|k} \end{bmatrix} = \begin{bmatrix} A & 0 \\ C & I \end{bmatrix} \begin{bmatrix} \hat{x}_{k+j|k} \\ \hat{I}_{k+j|k} \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} \hat{u}_{k+j|k} + \begin{bmatrix} 0 \\ -I \end{bmatrix} \hat{r}_{k+j|k} \quad (1.78a)$$

$$\hat{y}_{k+j|k} = \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} \hat{x}_{k+j|k} \\ \hat{I}_{k+j|k} \end{bmatrix} \quad (1.78b)$$

$$\hat{e}_{k+j|k} = \hat{y}_{k+j|k} - \hat{r}_{k+j|k} \quad (1.78c)$$

which is particularly simple when  $\hat{r}_{k+j|k} = 0$ . Hence, the model predictive controller with an integral term is implemented by estimating the states of the nominal model employing a Kalman filter. The integral state update is computed directly as  $I_{k+1} = I_k + (y_k - r_k)$ . The predictive controller employs prediction of the augmented model to compute the optimal controls. The main disadvantages of the applying integrating states to achieve offset free control is that extra tuning parameters are introduced and that an anti-windup algorithm may be needed to prevent an unnecessary performance penalty. This controller fits directly into the structure of the extended linear-quadratic optimal controller.

Prett and Garcia (1988), Muske and Rawlings (1993b), and Muske (1995) suggest a velocity form model predictive controller to achieve offset free control. They state the velocity model predictive controller for the deterministic system

$$x_{k+1} = Ax_k + Bu_k \quad (1.79a)$$

$$y_k = Cx_k \quad (1.79b)$$

as a controller based on the predictor

$$\begin{bmatrix} \Delta x_{k+1} \\ z_{k+1} \end{bmatrix} = \begin{bmatrix} A & 0 \\ CA & I \end{bmatrix} \begin{bmatrix} \Delta x_k \\ z_k \end{bmatrix} + \begin{bmatrix} B \\ CB \end{bmatrix} \Delta u_k \quad (1.80a)$$

$$y_k = \begin{bmatrix} 0 & I \end{bmatrix} \begin{bmatrix} \Delta x_k \\ z_k \end{bmatrix} \quad (1.80b)$$

and the objective function

$$\phi = \frac{1}{2} \sum_{k=0}^{N_p} (y_k - r_k)' Q (y_k - r_k) + \frac{1}{2} \sum_{k=0}^{N-1} \Delta u_k' S \Delta u_k \quad (1.81)$$

This controller has a similar structure to the model predictive controller with an integral error term and will obtain offset free control for systems with step disturbances. In this controller both the state estimator and the regulator are based on the augmented model. However, from a theoretical perspective this

construction is problematic as the velocity form of the equivalent stochastic model (1.73) would be

$$\begin{bmatrix} \Delta x_{k+1} \\ z_{k+1} \end{bmatrix} = \begin{bmatrix} A & 0 \\ CA & I \end{bmatrix} \begin{bmatrix} \Delta x_k \\ z_k \end{bmatrix} + \begin{bmatrix} B \\ CB \end{bmatrix} \Delta u_k + \begin{bmatrix} I & 0 \\ C & I \end{bmatrix} \begin{bmatrix} \Delta w_k \\ \Delta v_{k+1} \end{bmatrix} \quad (1.82a)$$

$$y_k = \begin{bmatrix} 0 & I \end{bmatrix} \begin{bmatrix} \Delta x_k \\ z_k \end{bmatrix} \quad (1.82b)$$

In this model, the process noise at sample  $k$  is certainly not independent from previous or future process noise. Consequently, the standard Kalman filter would not apply as an optimal estimator and this procedure should just be considered as an ad hoc methodology for achieving offset free model predictive control. Pannocchia and Rawlings (2001) list the advantages and disadvantage of velocity based model predictive control.

The other approach to eliminating steady-state offset involves augmenting the process model to include a constant step disturbance. The disturbance, which is estimated from the measured process variables, is generally assumed to remain constant. Closed-loop model predictive control performance is directly related to how accurately the disturbance model represents the actual disturbance entering the process (Francis and Wonham, 1976). Muske and Badgwell (2002) discuss offset free model predictive control by state augmentation using the *structured* disturbance model

$$\begin{bmatrix} x_{k+1} \\ d_{k+1} \\ p_{k+1} \end{bmatrix} = \begin{bmatrix} A & G_d & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} x_k \\ d_k \\ p_k \end{bmatrix} + \begin{bmatrix} B \\ 0 \\ 0 \end{bmatrix} u_k + \begin{bmatrix} w_k \\ \xi_k \\ \zeta_k \end{bmatrix} \quad (1.83a)$$

$$y_k = \begin{bmatrix} C & 0 & G_p \end{bmatrix} \begin{bmatrix} x_k \\ d_k \\ p_k \end{bmatrix} + v_k \quad (1.83b)$$

In this model,  $x_k$  is the states of the process model,  $d_k$  is used to model input and state disturbances, and  $p_k$  is used to model output disturbances. Similarly, Pannocchia and Rawlings (2003) discuss offset free model predictive control by disturbance estimation in the *unstructured* disturbance augmented model

$$\begin{bmatrix} x_{k+1} \\ d_{k+1} \end{bmatrix} = \begin{bmatrix} A & B_d \\ 0 & I \end{bmatrix} \begin{bmatrix} x_k \\ d_k \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u_k + \begin{bmatrix} w_k \\ \xi_k \end{bmatrix} \quad (1.84a)$$

$$y_k = \begin{bmatrix} C & C_d \end{bmatrix} \begin{bmatrix} x_k \\ d_k \end{bmatrix} + v_k \quad (1.84b)$$

One condition for offset free control by disturbance estimation is that the augmented model is detectable such that the Kalman-estimates of the process states as well as the disturbances approach the true states and disturbances. If the augmented system is not detectable, offset free control cannot be guaranteed in general. It should be noted that the results of Muske and Badgwell (2002) as

well as Pannocchia and Rawlings (2003) are derived for infinite-horizon predictive controllers with a target calculator. As the disturbance dynamics cannot be affected by the regulator, only the process model is used as predictor in the regulator part of the predictive controller and the disturbances are taken as constants. This situation is another case in which the extended linear quadratic regulator formulation (1.3) with linear terms in the objective function and affine terms in the state dynamics facilitates a transparent implementation.

The disturbance models may be identified as part of the process. However, it is often better to regard them as a controller device used to have offset free predictive control. The disadvantage of requiring offset free model predictive control is that the system may be destabilized in face of uncertain process models (c.f. Pannocchia and Rawlings, 2003). Furthermore, there is no general method to guarantee offset free inferential control unless the inferred variables are known deductively to be in the subspace of the measured variables.

### 1.1.5 Continuous-Time State Space Models

Predictive control based on nonlinear physical based models have appeared in recent years (Leineweber, 1999; Qin and Badgwell, 2000; Diehl *et al.*, 2001; Binder, 2002; Tenny, 2002). The application of nonlinear physical based models is facilitated by advances in modeling, experimental planning, and model identification as well as improved algorithms for estimation and predictive control of nonlinear models enabling real-time solutions. The success of each class of models for predictive control is to a large extent determined by the ease and state-of-art of modeling, experimental planning, identification, parameter estimation, state estimation, and control for a particular model class. For continuous-time systems integration software is needed for computation of the predictions as well. Therefore, the availability of such algorithms and software also influences the extent to which predictive control can be applied to these type of models (Young, 1981).

A large class of physical systems may be modeled deductively using systems of ordinary differential equations. The measurement of these systems is sampled at discrete-times such the system may be represented as

$$\dot{x}(t) = f(x(t), u(t), \theta) \quad (1.85a)$$

$$y_k = g(x_k, \theta) + e_k \quad (1.85b)$$

in which  $e_k$  may be interpreted as measurement noise as well residual due to system-model mismatch. The parameters in such systems are obtained using nonlinear regression (Bard, 1974; Seber and Wild, 1989), while the state estimation and predictive control may be employed using SQP algorithms for dynamic nonlinear optimization (Tenny, 2002).

Chemical systems which are partially governed by equilibrium processes are

more conveniently modeled deductively by DAE system of index one

$$\dot{x}(t) = f(x(t), z(t), u(t), \theta) \quad (1.86a)$$

$$0 = h(x(t), z(t), \theta) \quad (1.86b)$$

$$y_k = g(x_k, z_k, \theta) + e_k \quad (1.86c)$$

rather than ordinary differential equations. Software has appeared for parameter estimation and optimal control of index one DAE systems (Leineweber, 1999; Diehl *et al.*, 2001). In the ODE models as well as the DAE models the stochastics seems to be restricted to residuals,  $e_k$ , obtained when fitting the model to data.

To overcome such shortcomings, Melgaard (1994) and Kristensen (2002) suggested an approach in which the dynamic evolution of the system is described by stochastic ordinary differential equations

$$dx_t = f(x_t, u_t, \theta)dt + \sigma(u_t, \theta)dw_t \quad (1.87a)$$

$$y_k = g(x_k, u_k, \theta) + v_k \quad (1.87b)$$

Kristensen *et al.* (2002) provide software for parameter estimation in this class of models. This software is able to compute maximum a posteriori estimates of the parameters based on an extended Kalman filter predictor compatible with the continuous-discrete stochastic system (1.87). While the maximum a posteriori estimate based on the extended Kalman filter is not the true posteriori estimate of the stochastic system (1.87), Kristensen (2002) argue that it is a good approximation. In relation to predictive control, it is good prediction capabilities of the estimator and predictor that is of prime importance. The true system parameters are of less interest as long the as estimator has good prediction capabilities.

However, at present nonlinear stochastic predictive control is limited to being a research topic with challenges such as real-time computability. The parameter estimation in stochastic differential equations has been reviewed by Nielsen *et al.* (2000).

### 1.1.6 Neural Network Models

Fundamental continuous-time models tend to be very expensive to construct for most industrial processes. Therefore, in light of the success of linear empirical models for predictive control, it seems natural to consider black-box empirical models for nonlinear model predictive control (Sjöberg *et al.*, 1995; Lee, 2000). The empirical black box model may be denoted as

$$y_{k+1} = F(\phi_k, \theta) + e_k \quad (1.88)$$

in which  $y_k$  is the measured output at time index  $k$ ,  $\phi_k$  is a regressor at time  $k$ ,  $\theta$  is the parameter vector, and  $e_k$  is the residuals.  $\{e_k\}$  is usually assumed

to be white noise.  $F$  is the nonlinear empirical model. The regressor,  $\phi_k$ , may as in ARX-models consist of past measurements and inputs

$$\phi_k = [y'_k \ y'_{k-1} \ \dots \ y'_{k-n_y} \ u'_k \ u'_{k-1} \ \dots \ u'_{k-n_u}]' \quad (1.89a)$$

or of past measurements, inputs, and residuals

$$\phi_k = [y'_k \ \dots \ y'_{k-n_y} \ u'_k \ \dots \ u'_{k-n_u} \ e'_k \ \dots \ e'_{k-n_e}]' \quad (1.89b)$$

as in ARMAX models. The predictor given by these models is

$$\hat{y}_{k+1|k} = F(\phi_k, \theta) \quad (1.90a)$$

$$\hat{y}_{k+j+1|k} = F(\hat{\phi}_{k+j|k}, \theta) \quad j = 1, 2, \dots \quad (1.90b)$$

in which the predicted regression vector,  $\hat{\phi}_{k+j|k}$ , is given by the predicted measurements and inputs as well as past measurements and inputs. In example, for the NARMAX structure

$$\hat{\phi}_{k+j|k} = \begin{bmatrix} \hat{y}'_{k+j|k} & \dots & \hat{y}'_{k+j-n_y|k} & \hat{u}'_{k+j|k} & \dots & \hat{u}'_{k+j-n_u} \\ \hat{e}'_{k+j|k} & \dots & \hat{e}'_{k+j-n_e|k} \end{bmatrix}' \quad (1.91)$$

in which<sup>1</sup>

$$\hat{y}_{k+i|k} = y_{k+i} \quad i \leq 0 \quad (1.92a)$$

$$\hat{u}_{k+i|k} = u_{k+i} \quad i \leq 0 \quad (1.92b)$$

$$\hat{e}_{k+i|k} = \begin{cases} y_{k+i+1} - F(\phi_{k+i}, \theta) & i < 0 \\ 0 & i \geq 0 \end{cases} \quad (1.92c)$$

For the multilayer perceptron neural network with one hidden layer, the empirical nonlinear function,  $F(\phi_k, \theta) = [F_1(\phi_k, \theta) \ F_2(\phi_k, \theta) \ \dots \ F_p(\phi_k, \theta)]'$ , is given by (Nørgaard *et al.*, 2000)

$$x_j(\phi(k), \theta) = \chi_j \left( \sum_{t=1}^{n_\phi} w'_{jt} \phi_t(k) + w_{j0} \right) \quad j = 1, 2, \dots, n_h \quad (1.93a)$$

$$F_i(\phi(k), \theta) = \alpha_i \chi_i \left( \sum_{j=1}^{n_h} W_{ij} x_j(\phi(k), \theta) + W_{i0} \right) + \beta_i \quad i = 1, 2, \dots, p \quad (1.93b)$$

in which  $\phi(k) = \phi_k \in \mathbb{R}^{n_\phi}$ , the parameter vector  $\theta$  consists of the weights,  $w_{jt}$  and  $W_{ij}$ , as well as  $\alpha_i$  and  $\beta_i$ .  $n_h$  is the number of hidden layers. The basis

<sup>1</sup>This is an ad hoc predictor and not a general optimal predictor for (1.88). However, the idea is to estimate the parameters  $\theta$  such that the estimated structure is the predictor (1.91) rather than the model.

functions used to model the perceptrons in the neural network is usually the sigmoid function

$$\chi(x) = \frac{1}{1 + \exp(-x)} \quad (1.94)$$

or the hyperbolic tangent function

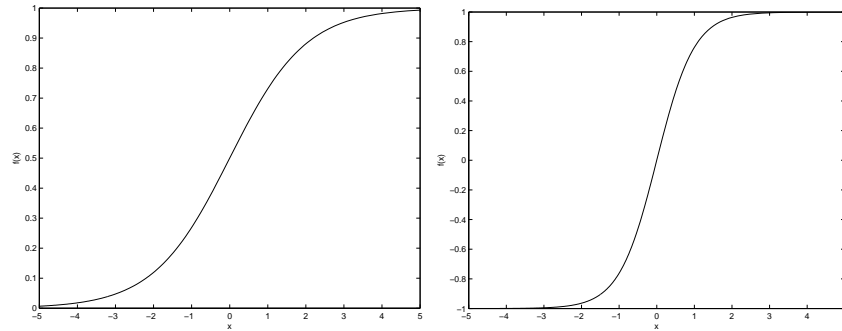
$$\chi(x) = \tanh(x) = \frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)} \quad (1.95)$$

The neural network model is able to provide excellent control when the operating space explored by the controller is within the operating space on which the neural network has been trained. However, in cases for which the controller drives the system to an operating space on which the model has not been trained, the neural network model generally has poor predictive performance.

Piché *et al.* (2000) have implemented and commercialized a neural network predictive controller different from the neural network predictive controller described above. Their predictive controller applies a neural network trained on historical data to predict the steady states. The prediction applied by the predictive controller is based on a linear input-output model gain-scheduled with the static neural network model (Qin and Badgwell, 2000).

## 1.2 Prediction Horizons and Stability

Stability issues have motivated formulation of model predictive controllers using an infinite horizon. Mayne *et al.* (2000) summarize the major nominally stabilizing formulations of model predictive control and show that the infinite horizon approximation is an important ingredient in these formulations. LaSalle



**Figure 1.1.** Some basis functions used in neural networks. Left:  $f(x) = 1/(1 + e^{-x})$ . Right:  $f(x) = \tanh(x)$

(1976, 1986) and Vidyasagar (1993) provide an introductory description of the stability issues and methods for stability analysis. Scokaert *et al.* (1997) provide a nice collection of discrete-time stability results of relevance to model predictive control.

In this section, the major state-space formulations of model predictive control for linear systems are summarized. The formulations with guaranteed nominal stability attempts to approximate the constrained *infinite* horizon linear quadratic regulation problem.

### 1.2.1 Finite-Horizon Optimal Regulation and Control

The finite horizon model predictive controller expressed by state space models may be expressed as (Bortolotto, 1985; Bortolotto and Jørgensen, 1986)

$$\min \quad \phi = \frac{1}{2} \sum_{k=0}^M x'_k Q x_k + u'_k R u_k + \frac{1}{2} \sum_{k=M+1}^N x'_k Q x_k \quad (1.96a)$$

$$s.t. \quad x_{k+1} = A x_k + B u_k \quad k = 0, 1, \dots, M \quad (1.96b)$$

$$C x_k + D u_k \geq b \quad k = 0, 1, \dots, M \quad (1.96c)$$

$$x_{k+1} = A x_k \quad k = M+1, M+2, \dots, N-1 \quad (1.96d)$$

$$C x_k \geq b \quad k = M+1, M+2, \dots, N \quad (1.96e)$$

This program is a finite-dimensional quadratic program and may be solved using standard software for quadratic programs. However, the application of this model predictive controller can destabilize stable systems if the prediction horizon,  $N$  ( $N > M$ ), is too short.

### 1.2.2 Infinite-Horizon Optimal Regulation

Kalman (1960a) demonstrated that optimality is not sufficient for stability. Rather, under certain conditions (stabilizability and detectability) infinite horizon optimal controllers are stabilizing. An appropriate Lyapunov function for establishing stability is the value function associated with the infinite horizon optimal control problem. The infinite horizon linear quadratic optimal control problem

$$\min \quad \phi = \frac{1}{2} \sum_{k=0}^{\infty} x'_k Q x_k + u'_k R u_k \quad (1.97a)$$

$$s.t. \quad x_{k+1} = A x_k + B u_k \quad k = 0, 1, \dots \quad (1.97b)$$

has the optimal solution

$$u_k = K x_k \quad (1.98)$$



in which the gain  $K$  is computed using the Riccati equation (Bittani *et al.*, 1991)

$$\Pi = Q + A'\Pi A - A'\Pi B(R + B'\Pi B)^{-1}B'\Pi A \quad (1.99a)$$

$$K = -(R + B'\Pi B)^{-1}B'\Pi A \quad (1.99b)$$

The optimal value of (1.97) is

$$\begin{aligned} V(x_0) &= \left\{ \frac{1}{2} \sum_{k=0}^{\infty} x_k' (Q + K'RK) x_k : x_{k+1} = (A + BK)x_k \right\} \\ &= \frac{1}{2} x_0' \Pi x_0 \end{aligned} \quad (1.100)$$

The stabilizing property of the infinite horizon linear quadratic optimal controller has motivated study of an infinite horizon constrained linear quadratic optimal controller

$$\min \quad \phi = \frac{1}{2} \sum_{k=0}^{\infty} x_k' Q x_k + u_k' R u_k \quad (1.101a)$$

$$s.t. \quad x_{k+1} = Ax_k + Bu_k \quad k = 0, 1, \dots \quad (1.101b)$$

$$Cx_k + Du_k \geq b \quad k = 0, 1, \dots \quad (1.101c)$$

This controller is stabilizing under certain conditions (Mayne *et al.*, 2000). However, (1.101) is an infinite dimensional quadratic program which is not amenable to direct numerical solution. To be numerically solvable it must be transformed to an equivalent finite dimensional quadratic program. The transformation of the infinite stage objective function to a finite state objective function is facilitated by the solution of the unconstrained linear quadratic optimal control problem. The infinite number of constraints may be represented by a finite number of constraints using the theory of maximal output admissible sets developed by Gilbert and Tan (1991).

The maximal output admissible set defined as

$$\mathbb{O}_{\infty}(A, C, b) = \{x_0 \in \mathbb{R}^n \mid x_{k+1} = Ax_k, Cx_k \geq b, k = 0, 1, \dots\} \quad (1.102)$$

may be represented by a finite set of constraints (Gilbert and Tan, 1991)

$$\begin{aligned} \mathbb{O}_{\infty}(A, C, b) &= \mathbb{O}_t(A, C, b) \\ &= \{x_0 \in \mathbb{R}^n \mid x_{k+1} = Ax_k, Cx_k \geq b, k = 0, 1, \dots, t\} \end{aligned} \quad (1.103)$$

if origo is in the interior of  $\mathbb{Y} = \{x \in \mathbb{R}^n : Cx \geq b\}$ ,  $\mathbb{Y}$  is bounded,  $A$  is stable, and  $(A, C)$  is observable. The maximal output admissible set and the theory for its construction facilitates a finite representation of an infinite number of constraints. It may therefore be applied in construction of an approximate numerical solution of the infinite horizon constrained linear quadratic optimal

controller (1.101). The different approximations of the infinite horizon constrained linear quadratic optimal controller differ in the way they parameterize the tail,  $\{x_{k+1}, u_k\}_{k=N}^{\infty}$ , of the state-control trajectory  $\{x_k, u_k\}_{k=0}^{\infty}$ . This influences how the cost-to-go

$$\frac{1}{2} \sum_{k=N}^{\infty} x'_k Q x_k + u'_k R u_k$$

of the objective function

$$\begin{aligned} \phi &= \frac{1}{2} \sum_{k=0}^{\infty} x'_k Q x_k + u'_k R u_k \\ &= \frac{1}{2} \sum_{k=0}^{N-1} x'_k Q x_k + u'_k R u_k + \frac{1}{2} \sum_{k=N}^{\infty} x'_k Q x_k + u'_k R u_k \end{aligned}$$

is computed. The chosen parameterization also affects the representation of the maximal output admissible set. Different parameterizations that have appeared in the literature are presented in the following sections.

### 1.2.2.1 MPC with a Terminal Equality Constraint

Keerthi (1986) and Keerthi and Gilbert (1988) constructed an approximate solution to the infinite horizon constrained linear quadratic regulation problem (1.97) by employing the parameterization

$$x_N = 0 \tag{1.104a}$$

$$u_k = 0 \quad k = N, N+1, \dots \tag{1.104b}$$

In this parameterization, the endpoint,  $x_N$ , is fixed at the steady state. This implies that the states at all subsequent time points remain at the steady state

$$x_k = 0 \quad k = N+1, N+2, \dots \tag{1.105}$$

With this parameterization, the infinite horizon linear quadratic regulation problem (1.97) may be formulated as the finite dimensional quadratic program

$$\min_{\{x_{k+1}, u_k\}_{k=0}^{N-1}} \quad \phi = \frac{1}{2} \sum_{k=0}^{N-1} x'_k Q x_k + u'_k R u_k \tag{1.106a}$$

$$s.t. \quad x_{k+1} = Ax_k + Bu_k \quad k = 0, 1, \dots, N-1 \tag{1.106b}$$

$$Cx_k + Du_k \geq b \quad k = 0, 1, \dots, N-1 \tag{1.106c}$$

$$x_N = 0 \tag{1.106d}$$

In contrast to the infinite horizon constrained linear quadratic regulation problem (1.97), the endpoint constrained model predictive control problem is computable. Furthermore, this predictive controller is nominally stabilizing if a feasible solution exists (Keerthi and Gilbert, 1988). A feasible solution exists, provided that the horizon,  $N$ , is selected sufficiently long and a feasible solution exists to the infinite horizon constrained linear quadratic regulation problem exists.

The feasibility issue is also a major bottleneck for the implementation of the end-point constrained model predictive controller (1.106). For a given  $N$ , the problem (1.106) may be infeasible and the controller must be equipped with the appropriate logic to handle such situations and increase  $N$  or relax the problem formulation.

The main advantage of the predictive controller (1.106) is its conceptual and computational simple construction. Opposed to other infinite horizon predictive controllers, neither the maximal output admissible set nor the cost-to-go matrix need to be computed. As  $N \rightarrow \infty$  the solution of (1.106) approaches the solution of the infinite horizon constrained linear quadratic regulation problem (1.97). The simple formulation of the predictive controller (1.106) comes at the price of losing the optimal control problem structure. (1.106) is not a standard optimal control problem. Rather, it is a two-point boundary value problem.

### 1.2.2.2 Muske-Rawlings Parameterization

For stable systems, Muske and Rawlings (1993a) applied the parameterization

$$u_k = 0 \quad k = N, N+1, \dots \quad (1.107)$$

such that the infinite horizon constrained linear quadratic regulation problem (1.101) can be formulated as

$$\min \quad \phi = \frac{1}{2} \left( \sum_{k=0}^{N-1} x_k' Q x_k + u_k' R u_k \right) + \frac{1}{2} \sum_{k=N}^{\infty} x_k' Q x_k \quad (1.108a)$$

$$s.t. \quad x_{k+1} = A x_k + B u_k \quad k = 0, 1, \dots, N-1 \quad (1.108b)$$

$$x_{k+1} = A x_k \quad k = N, N+1, \dots \quad (1.108c)$$

$$C x_k + D u_k \geq b \quad k = 0, 1, \dots, N-1 \quad (1.108d)$$

$$C x_k \geq b \quad k = N, N+1, \dots \quad (1.108e)$$

The end predictions,  $x_{k+1} = A x_k$  for  $k \geq N$ , implies  $x_k = A^{k-N} x_N$  for  $k \geq N$  such that

$$\sum_{k=N}^{\infty} x_k' Q x_k = x_N' \left( \sum_{k=N}^{\infty} (A^{k-N})' Q A^{k-N} \right) x_N = x_N' P x_N \quad (1.109)$$

in which  $P$  may be computed from the Lyapunov equation (Gajic and Qureshi, 1995)

$$P = \sum_{k=N}^{\infty} (A')^{k-N} Q A^{k-N} = \sum_{j=0}^{\infty} (A')^j Q A^j = Q + A' P A \quad (1.110)$$

Consequently, for stable systems, the objective function of (1.101) may be represented as a finite sum. Similarly, the constraints  $x_{k+1} = Ax_k$  and  $Cx_k \geq b$  for  $k \geq N$  is equivalent with the maximal output admissible set,  $\mathbb{O}_{\infty}(A, C, b) = \mathbb{O}_t(A, C, b)$  such that the constrained infinite horizon linear quadratic regulation problem (1.108) may be expressed as a finite dimensional quadratic program

$$\min_{\{x_{k+1}, u_k\}_{k=0}^{N-1}} \phi = \frac{1}{2} \left( \sum_{k=0}^{N-1} x'_k Q x_k + u'_k R u_k \right) + \frac{1}{2} x'_N P x_N \quad (1.111a)$$

$$s.t. \quad x_{k+1} = Ax_k + Bu_k \quad k = 0, 1, \dots, N-1 \quad (1.111b)$$

$$Cx_k + Du_k \geq b \quad k = 0, 1, \dots, N-1 \quad (1.111c)$$

$$x_N \in \mathbb{O}_{\infty}(A, C, b) \quad (1.111d)$$

The solution of the infinite horizon linear quadratic optimal control problem may for stable systems be approximated by solution of (1.108) which is obtained by solution of the finite dimensional quadratic program (1.111).

For unstable systems, the infinite horizon controller using the parameterization  $u_k = 0$  for  $k \geq N$  must be modified slightly. A *real Schur decomposition* of the state transition matrix,  $A$ , may be used to decompose the system into a stable and an unstable subsystem. Let the real Schur decomposition of  $A$  be

$$Z' A Z = \begin{bmatrix} \bar{A}_s & \bar{A}_{12} \\ 0 & \bar{A}_u \end{bmatrix} \quad Z = \begin{bmatrix} Z_s & Z_u \end{bmatrix} \quad (1.112)$$

in which  $Z$  is an orthogonal matrix produced in the real Schur decomposition of  $A$  (Golub and Van Loan, 1996). Using the eigenvalues produced by the real Schur decomposition, the matrices resulting from the Schur decomposition is partitioned such that one block is associated with the stable subsystem and another block is associated with the unstable subsystem.

Using the similar transformation  $x_k = Z z_k$  and the convention  $z_k = \begin{bmatrix} z_k^s \\ z_k^u \end{bmatrix}$  implies that the system dynamics may be expressed as

$$\begin{bmatrix} z_{k+1}^s \\ z_{k+1}^u \end{bmatrix} = \begin{bmatrix} \bar{A}_s & \bar{A}_{12} \\ 0 & \bar{A}_u \end{bmatrix} \begin{bmatrix} z_k^s \\ z_k^u \end{bmatrix} + \begin{bmatrix} \bar{B}_s \\ \bar{B}_u \end{bmatrix} u_k \quad (1.113)$$

in which

$$\begin{bmatrix} \bar{A}_s & \bar{A}_{12} \\ 0 & \bar{A}_u \end{bmatrix} = Z' A Z \quad \begin{bmatrix} \bar{B}_s \\ \bar{B}_u \end{bmatrix} = Z' B \quad (1.114)$$

The transformed state space system along with the parameterization

$$z_N^u = 0 \quad (1.115a)$$

$$u_k = 0 \quad k = N, N+1, \dots \quad (1.115b)$$

implies  $z_k^u = 0$  for  $k \geq N$ . This parameterization is called the Muske-Rawlings parameterization for unstable systems. With this parameterization, the approximation of the infinite-horizon constrained linear quadratic regulation problem becomes

$$\min \phi = \frac{1}{2} \sum_{k=0}^{N-1} \begin{bmatrix} z_k^s \\ z_k^u \end{bmatrix}' \begin{bmatrix} \bar{Q}_{ss} & \bar{Q}_{su} \\ \bar{Q}_{us} & \bar{Q}_{uu} \end{bmatrix} \begin{bmatrix} z_k^s \\ z_k^u \end{bmatrix} + u_k' R u_k + \frac{1}{2} \sum_{k=N}^{\infty} (z_k^s)' \bar{Q}_{ss} z_k^s \quad (1.116a)$$

$$s.t. \quad \begin{bmatrix} z_{k+1}^s \\ z_{k+1}^u \end{bmatrix} = \begin{bmatrix} \bar{A}_s & \bar{A}_{12} \\ 0 & \bar{A}_u \end{bmatrix} \begin{bmatrix} z_k^s \\ z_k^u \end{bmatrix} + \begin{bmatrix} \bar{B}_s \\ \bar{B}_u \end{bmatrix} u_k \quad k = 0, 1, \dots, N-1 \quad (1.116b)$$

$$z_{k+1}^s = \bar{A}_s z_k^s \quad k = N, N+1, \dots \quad (1.116c)$$

$$\begin{bmatrix} \bar{C}_s & \bar{C}_u \end{bmatrix} \begin{bmatrix} z_k^s \\ z_k^u \end{bmatrix} + D u_k \geq b \quad k = 0, 1, \dots, N-1 \quad (1.116d)$$

$$\bar{C}_s z_k^s \geq b \quad k = N, N+1, \dots \quad (1.116e)$$

$$z_N^u = 0 \quad (1.116f)$$

in which

$$\begin{bmatrix} \bar{C}_s & \bar{C}_u \end{bmatrix} = C Z \quad \begin{bmatrix} \bar{Q}_{ss} & \bar{Q}_{su} \\ \bar{Q}_{us} & \bar{Q}_{uu} \end{bmatrix} = Z' Q Z \quad (1.117)$$

Assume that origo is in the interior of the set  $\{\bar{C}_s z_k^s \geq b\}$ , i.e. that  $b < 0$ . Then the infinite dimensional mathematical program (1.116) may be represented as a finite dimensional mathematical program

$$\min \quad \phi = \frac{1}{2} \sum_{k=0}^{N-1} x_k' Q x_k + u_k' R u_k + \frac{1}{2} x_N' P x_N \quad (1.118a)$$

$$s.t. \quad x_{k+1} = A x_k + B u_k \quad k = 0, 1, \dots, N-1 \quad (1.118b)$$

$$C x_k + D u_k \geq b \quad k = 0, 1, \dots, N-1 \quad (1.118c)$$

$$Z_s' x_N \in \mathbb{O}_{\infty}(Z_s' A Z_s, C Z_s, b) \quad (1.118d)$$

$$Z_u' x_N = 0 \quad (1.118e)$$

in which

$$P = \begin{bmatrix} \bar{P} & 0 \\ 0 & 0 \end{bmatrix} \quad \bar{P} = \bar{Q}_{ss} + \bar{A}_s' \bar{P} \bar{A}_s \quad (1.119)$$

$\bar{P}$  is computed from the Lyapunov equation associated with the stable subspace.

The Muske-Rawlings parameterized model predictive controller for linear stable and unstable systems are nominally stabilizing (Rawlings and Muske, 1993; Muske and Rawlings, 1993a,b). The Muske-Rawlings parameterization for unstable systems may be thought of as a combination of the terminal constrained MPC and the Muske-Rawlings MPC for stable systems.

### 1.2.2.3 Dual Mode Parameterization

Another and often applied parameterization of the tail trajectory is

$$u_k = Lx_k \quad k = N, N + 1, \dots \quad (1.120)$$

in which  $L$  is some matrix that stabilizes  $x_{k+1} = Ax_k + Bu_k$ . In this case, the constrained optimal control problem may be expressed by the infinite dimensional mathematical program

$$\min \quad \phi = \frac{1}{2} \left( \sum_{k=0}^{N-1} x'_k Q x_k + u'_k R u_k \right) + \frac{1}{2} \sum_{k=N}^{\infty} x'_k (Q + L' R L) x_k \quad (1.121a)$$

$$s.t. \quad x_{k+1} = Ax_k + Bu_k \quad k = 0, 1, \dots, N-1 \quad (1.121b)$$

$$x_{k+1} = (A + BL)x_k \quad k = N, N + 1, \dots \quad (1.121c)$$

$$Cx_k + Du_k \geq b \quad k = 0, 1, \dots, N-1 \quad (1.121d)$$

$$(C + DL)x_k \geq b \quad k = N, N + 1, \dots \quad (1.121e)$$

The maximal output admissible set

$$\begin{aligned} \mathbb{O}_{\infty}((A + BL), (C + DL), b) = \\ \{x_N \in \mathbb{R}^n : x_{k+1} = (A + BL)x_k, (C + DL)x_k \geq b, k = N, N + 1, \dots\} \end{aligned} \quad (1.122)$$

may be represented by a finite set of constraints (Gilbert and Tan, 1991). The solution,  $P$ , of the Lyapunov equation

$$P = (Q + L' R L) + (A + BL)' P (A + BL) \quad (1.123)$$

can be used to compute the cost-to-go as

$$\left\{ \sum_{k=N}^{\infty} x'_k (Q + L' R L) x_k : x_{k+1} = (A + BL)x_k \right\} = x'_N P x_N \quad (1.124)$$

Consequently, the infinite dimensional program (1.121) may be represented by the equivalent finite dimensional quadratic program

$$\min \quad \phi = \frac{1}{2} \left( \sum_{k=0}^{N-1} x'_k Q x_k + u'_k R u_k \right) + \frac{1}{2} x'_N P x_N \quad (1.125a)$$

$$s.t. \quad x_{k+1} = Ax_k + Bu_k \quad k = 0, 1, \dots, N-1 \quad (1.125b)$$

$$Cx_k + Du_k \geq b \quad k = 0, 1, \dots, N-1 \quad (1.125c)$$

$$x_N \in \mathbb{O}_{\infty}((A + BL), (C + DL), b) \quad (1.125d)$$

As (1.125) is finite dimensional it can be solved using quadratic programming solution techniques. The parameterization (1.120) of (1.121) has been referred

to as a dual-mode controller (Mayne and Michalska, 1990). A special and important case of the dual mode parameterization is to select the matrix  $L$  as the solution,  $K$ , of the LQR problem, i.e.

$$u_k = Kx_k \quad (1.126)$$

With this parameterization, the infinite horizon linear quadratic regulation problem (1.101) may be expressed as

$$\min \quad \phi = \frac{1}{2} \left( \sum_{k=0}^{N-1} x'_k Q x_k + u'_k R u_k \right) + \frac{1}{2} \sum_{k=N}^{\infty} x'_k (Q + K' R K) x_k \quad (1.127a)$$

$$s.t. \quad x_{k+1} = Ax_k + Bu_k \quad k = 0, 1, \dots, N-1 \quad (1.127b)$$

$$x_{k+1} = (A + BK)x_k \quad k = N, N+1, \dots \quad (1.127c)$$

$$Cx_k + Du_k \geq b \quad k = 0, 1, \dots, N-1 \quad (1.127d)$$

$$(C + DK)x_k \geq b \quad k = N, N+1, \dots \quad (1.127e)$$

and converted to a finite dimensional quadratic program

$$\min \quad \phi = \frac{1}{2} \left( \sum_{k=0}^{N-1} x'_k Q x_k + u'_k R u_k \right) + \frac{1}{2} x'_N \Pi x_N \quad (1.128a)$$

$$s.t. \quad x_{k+1} = Ax_k + Bu_k \quad k = 0, 1, \dots, N-1 \quad (1.128b)$$

$$Cx_k + Du_k \geq b \quad k = 0, 1, \dots, N-1 \quad (1.128c)$$

$$x_N \in \mathbb{O}_{\infty}((A + BK), (C + DK), b) \quad (1.128d)$$

using the maximal output admissible set and the fact that the cost-to-go matrix,  $P$ , in this special case equals the solution,  $\Pi$ , of the Riccati equation (1.99a) due to the selection of  $K$ .

#### 1.2.2.4 Constrained Linear Quadratic Regulation

The solutions of the finite dimensional mathematical programs (1.111), (1.125), and (1.128) are all approximations to the solution of the infinite horizon constrained linear quadratic regulation problem (1.101). Chmielewski and Manou-siouthakis (1996) and Scokaert and Rawlings (1998) independently developed numerical solution procedures for the exact solution of the infinite horizon constrained linear quadratic regulation problem (1.101). The procedure of Scokaert and Rawlings (1998) for solution of (1.101) requires construction of the maximal output admissible set  $\mathbb{O}_{\infty}((A + BK), (C + DK), b)$  in which  $K$  is the gain computed as part of the solution of the unconstrained LQR problem (1.97). As the procedure requires the maximal output admissible set  $\mathbb{O}_{\infty}((A + BK), (C + DK), b)$  this set must be computable. As explained by Gilbert and Tan (1991) this set is computable if origo is in the interior of  $\mathbb{Y} = \{x \in \mathbb{R}^n : (C + DK)x \geq b\}$ ,  $\mathbb{Y}$  is bounded,  $A + BK$  is stable, and

$((C + DK), (A + BK))$  is observable. Under these assumptions, Scokaert and Rawlings (1998) noted that if  $x_0 \in \mathbb{O}_\infty((A + BK), (C + DK), b)$  then the solution of the infinite horizon *constrained* linear quadratic regulation problem (1.101) is identical to the solution of the infinite horizon *unconstrained* linear quadratic regulation problem (1.97). Consequently, the solution of the finite dimensional mathematical program

$$\min \quad \phi = \frac{1}{2} \left( \sum_{k=0}^{N-1} x_k' Q x_k + u_k' R u_k \right) + \frac{1}{2} x_N' \Pi x_N \quad (1.129a)$$

$$s.t. \quad x_{k+1} = A x_k + B u_k \quad k = 0, 1, \dots, N-1 \quad (1.129b)$$

$$C x_k + D u_k \geq b \quad k = 0, 1, \dots, N-1 \quad (1.129c)$$

with the implicit tail parameterization

$$u_k = K x_k \quad k = N, N+1, \dots \quad (1.130)$$

is identical to the solution of the infinite horizon constrained linear quadratic regulation problem (1.101) provided  $x_N^* \in \mathbb{O}_\infty((A + BK), (C + DK), b)$ . Furthermore, Scokaert and Rawlings (1998) proved that there exists a finite  $N$  such that  $x_N^* \in \mathbb{O}_\infty((A + BK), (C + DK), b)$  provided origo is in the interior of the maximal output admissible set. By these observations, the infinite horizon constrained linear quadratic regulation problem (1.101) may be solved by the procedure in the following proposition

**Proposition 1.2.1 (Constrained LQR Solution Algorithm)**

1. Compute  $\Pi$  and  $K$  using (1.99).
2. Construct a finite representation of  $\mathbb{O}_\infty((A + BK), (C + DK), b)$  using the algorithm of Gilbert and Tan (1991).
3. Choose a finite horizon  $N_0$ . Set  $N = N_0$ .
4. Solve (1.129). Denote the corresponding optimal solution  $\pi_N^*$ .
5. If  $x_N^* \in \mathbb{O}_\infty((A + BK), (C + DK), b)$  then stop. The optimal solution of (1.101) is  $\pi_N^*$ .
6. Increase  $N$  and go to step 4.

**1.2.2.5 Active Steady State Constraints**

The algorithm proposed by Scokaert and Rawlings (1998) presumes that origo is in the interior of the feasible region,  $\mathbb{Y}$ . This assumption precludes systems with constraints active at steady state. When state constraints are active at steady state, arbitrarily small constant disturbances would render the hard constrained problem infeasible, which means that there is no feasible sequence that brings the system to the origin without persistently violating the active constraints permanently. Therefore it seems reasonable by definition of the control problem to assume that state constraints are not active at steady state.



However, input constraints are frequently active at steady state in order to reject large disturbances and maximize production. Rao and Rawlings (1999) constructed suboptimal solutions for situations with active steady state constraints. The inputs at a steady state constraint were fixed at their bounds in computation of the cost-to-go and associated gain matrix. Pannocchia *et al.* (2002) provided a rigorous solution of the infinite horizon constrained linear quadratic optimal regulation problem without the assumption that the steady inputs are in the interior of the feasible region. This implies that they can handle the situation with active input constraints rigorously. General constraints are still not allowed to be active at steady state. Pannocchia *et al.* (2002) developed their algorithm for the following infinite horizon constrained linear quadratic regulation problem

$$\min \quad \phi = \frac{1}{2} \sum_{k=0}^{\infty} x_k' Q x_k + u_k' R u_k \quad (1.131a)$$

$$s.t. \quad x_{k+1} = A x_k + B u_k \quad k = 0, 1, \dots \quad (1.131b)$$

$$C x_k + D u_k \geq b \quad k = 0, 1, \dots \quad (1.131c)$$

$$u_k \geq 0 \quad k = 0, 1, \dots \quad (1.131d)$$

This program includes an explicit representation (1.131d) of input constraints. Denote the feasible region of (1.131) as  $\Omega$ . An upper bound,  $\phi_u^*(N) \geq \phi^*$ , of the value function of (1.131) may be generated by solution of

$$\min \quad \phi_u(N) = \frac{1}{2} \sum_{k=0}^{\infty} x_k' Q x_k + u_k' R u_k \quad (1.132a)$$

$$s.t. \quad x_{k+1} = A x_k + B u_k \quad (1.132b)$$

$$C x_k + D u_k \geq b \quad (1.132c)$$

$$u_k \geq 0 \quad (1.132d)$$

$$E u_k = 0 \quad k = N, N+1, \dots \quad (1.132e)$$

as the feasible region,  $\Omega_u(N)$ , of (1.132) is a subset of the feasible region,  $\Omega$ , of (1.131). The matrix  $E$  is selected such that all inputs whose bounds are active at steady state are zeroed. The infinite-dimensional mathematical program

(1.132) may be formulated as

$$\min \quad \phi_u(N) = \frac{1}{2} \sum_{k=0}^{N-1} x'_k Q x_k + u'_k R u_k + \frac{1}{2} \sum_{k=N}^{\infty} x'_k Q x_k + \bar{u}'_k \bar{R} \bar{u}_k \quad (1.133a)$$

$$s.t. \quad x_{k+1} = A x_k + B u_k \quad k = 0, 1, \dots, N-1 \quad (1.133b)$$

$$x_{k+1} = A x_k + \bar{B} \bar{u}_k \quad k = N, N+1, \dots \quad (1.133c)$$

$$C x_k + D u_k \geq b \quad k = 0, 1, \dots, N-1 \quad (1.133d)$$

$$C x_k + \bar{D} \bar{u}_k \geq b \quad k = N, N+1, \dots \quad (1.133e)$$

$$u_k \geq 0 \quad k = 0, 1, \dots, N-1 \quad (1.133f)$$

$$\bar{u}_k \geq 0 \quad k = N, N+1, \dots \quad (1.133g)$$

in which the barred quantities are associated with the inputs whose bounds are not active at steady state. This program may be solved using the algorithm of Scokaert and Rawlings (1998) for constrained linear quadratic regulation problems with origo in the interior of the feasible region. By construction of  $\bar{u}_k$  and by assumption with respect to the constraints  $C x_k + D u_k \geq b$ , origo is in the interior of the feasible region of (1.133) such that the prerequisites for applying the Scokaert-Rawlings algorithm are indeed met. A conceptual alternative for solution of (1.132) is presented by Rao and Rawlings (1999).

A lower bound,  $\phi_l^*(N) \leq \phi^*$ , of the value function of (1.131) may be obtained by solution of

$$\min \quad \phi_l(N) = \frac{1}{2} \sum_{k=0}^{\infty} x'_k Q x_k + u'_k R u_k \quad (1.134a)$$

$$s.t. \quad x_{k+1} = A x_k + B u_k \quad k = 0, 1, \dots \quad (1.134b)$$

$$C x_k + D u_k \geq b \quad k = 0, 1, \dots, N-1 \quad (1.134c)$$

$$u_k \geq 0 \quad k = 0, 1, \dots, N-1 \quad (1.134d)$$

as the feasible region,  $\Omega_l(N)$ , of (1.134) is related to the feasible region,  $\Omega$ , of (1.131) by:  $\Omega_l(N) \supset \Omega$ . Numerically, the solution of (1.134) is obtained by solution of the equivalent finite dimensional quadratic program

$$\min \quad \phi_l(N) = \frac{1}{2} \sum_{k=0}^{N-1} x'_k Q x_k + u'_k R u_k + \frac{1}{2} x'_N \Pi x_N \quad (1.135a)$$

$$s.t. \quad x_{k+1} = A x_k + B u_k \quad k = 0, 1, \dots, N-1 \quad (1.135b)$$

$$C x_k + D u_k \geq b \quad k = 0, 1, \dots, N-1 \quad (1.135c)$$

$$u_k \geq 0 \quad k = 0, 1, \dots, N-1 \quad (1.135d)$$

in which  $\Pi$  is computed as the solution of the discrete-time Riccati equation (1.99a). The sequence  $\{\phi_l^*(N)\}$  is a non-decreasing sequence that approaches  $\phi^*$  as  $N \rightarrow \infty$ . Similarly, the sequence  $\{\phi_u^*(N)\}$  is a non-increasing sequence

that approaches  $\phi^*$  as  $N \rightarrow \infty$ . The convergence of  $\phi_l^*(N)$  and  $\phi_u^*(N)$  to  $\phi^*$  and the convexity of the involved optimization problems are used to establish a guaranteed accuracy on the input signal  $u_0^*(N)$  computed by solution of (1.132). The accuracy of  $u_0^*(N)$  compared to the optimal solution  $u_0^*$  is given by the expression

$$\|u_0^*(N) - u_0^*\| \leq \frac{2}{\alpha}(\phi_u^*(N) - \phi_l^*(N)) \quad (1.136)$$

in which  $\alpha$  is a positive constant:  $\alpha \geq \lambda_{\min}(R) > 0$ .

### 1.2.3 Transcription to the Regulation Problem

Most process control applications are not only regulation problems but in addition to being regulation problems also either target tracking problems or disturbance rejection problems. In the target tracking problem, the actuators  $\{u_k\}$  are computed such that a given reference signal  $\{r_k\}$  is tracked. In the disturbance rejection problem, the actuators  $\{u_k\}$  is computed to compensate for estimated or measured disturbances  $\{d_k\}$ . The target tracking problem as well as the disturbance rejection problem contain the regulation problem as well. Qualitatively, the regulation problem can be regarded as selection of the actuator values  $\{u_k\}$  such that a perturbed state  $x_0$  is brought back to origo. The general infinite-horizon linear-quadratic optimal control problem containing the regulation problem, the target tracking problem and the disturbance rejection problem may be formulated as

$$\min \quad \phi = \frac{1}{2} \sum_{k=0}^{\infty} (y_k - r_k)' Q_y (y_k - r_k) + (u_k - \bar{u})' R (u_k - \bar{u}) \quad (1.137a)$$

$$s.t. \quad x_{k+1} = Ax_k + Bu_k + Ed_k \quad (1.137b)$$

$$y_k = Cx_k \quad (1.137c)$$

$$Fx_k + Gu_k + Hd_k \geq b \quad (1.137d)$$

in which  $\{r_k\}$  is a specified set of output targets and  $\{d_k\}$  is a given estimated sequence of disturbances. It is assumed that the the output reference trajectory reaches some constant value,  $r_s$ , and that the disturbance trajectory reaches some constant value,  $d_s$

$$\{r_k\} = \{r_0, r_1, \dots, r_{N-1}, r_s, r_s, \dots\} \quad (1.138a)$$

$$\{d_k\} = \{d_0, d_1, \dots, d_{N-1}, d_s, d_s, \dots\} \quad (1.138b)$$

The stage costs of the objective function are also denoted

$$g_k(y_k, u_k) = \frac{1}{2} (y_k - r_k)' Q_y (y_k - r_k) + (u_k - \bar{u})' R (u_k - \bar{u}) \quad (1.139)$$

such that we may express the objective function as

$$\phi = \sum_{k=0}^{\infty} g_k(y_k, u_k) \quad (1.140)$$

The solution of the general infinite-horizon linear quadratic optimal control problem (1.137) is solved by decomposing the problem into a steady-state problem and a regulation problem (Muske and Rawlings, 1993a; Muske, 1997; Rao and Rawlings, 1999; Jørgensen and Rawlings, 2000). The steady-state problem of (1.137) may be formulated as (Muske, 1997)

$$\min_{u_s, x_s, y_s} \quad g_s(y_s, u_s) = \frac{1}{2}(y_s - r_s)'Q_y(y_s - r_s) + \frac{1}{2}(u_s - \bar{u})'R(u_s - \bar{u}) \quad (1.141a)$$

$$s.t. \quad x_s = Ax_s + Bu_s + Ed_s \quad (1.141b)$$

$$y_s = Cx_s \quad (1.141c)$$

$$Fx_s + Gu_s + Hd_s \geq b \quad (1.141d)$$

$r_s$  and  $d_s$  are the given steady state output reference and disturbance, respectively. Given the optimal steady state values  $(u_s, x_s, y_s)$  and the steady state disturbance,  $d_s$ , the deviation variables may be defined as

$$w_k = x_k - x_s \quad (1.142a)$$

$$v_k = u_k - u_s \quad (1.142b)$$

$$z_k = y_k - y_s \quad (1.142c)$$

$$\xi_k = d_k - d_s \quad (1.142d)$$

Define the quantities  $\bar{q}_k$  and  $s_k$  as

$$\bar{q}_k = Q_y(y_s - r_k) \quad (1.143a)$$

$$s_k = R(u_s - \bar{u}) \quad (1.143b)$$

and observe that they are completely known when the steady state problem (1.141) has been solved. Using the deviation variables as well as  $\bar{q}_k$  and  $s_k$ , the objective function,  $\phi$ , of (1.137) may be expressed as

$$\begin{aligned} \phi &= \frac{1}{2} \sum_{k=0}^{\infty} (y_k - r_k)'Q_y(y_k - r_k) + (u_k - \bar{u})'R(u_k - \bar{u}) \\ &= \frac{1}{2} \sum_{k=0}^{\infty} [(y_k - y_s) + (y_s - r_k)]'Q_y[(y_k - y_s) + (y_s - r_k)] \\ &\quad + [(u_k - u_s) + (u_s - \bar{u})]'R[(u_k - u_s) + (u_s - \bar{u})] \\ &= \frac{1}{2} \sum_{k=0}^{\infty} [z_k'Q_y z_k + v_k'R v_k + 2\bar{q}_k' z_k + 2s_k' v_k \\ &\quad + (y_s - r_k)'Q_y(y_s - r_k) + (u_s - \bar{u})'R(u_s - \bar{u})] \\ &= \sum_{k=0}^{\infty} \left[ \frac{1}{2} (z_k'Q_y z_k + v_k'R v_k + 2\bar{q}_k' z_k + 2s_k' v_k) + g_s(y_s, u_s) \right] \end{aligned} \quad (1.144)$$

This objective function may be infinite for all feasible control trajectories. However, by subtracting the optimal steady-state costs it is conjectured that it can

be made finite at least for the optimal solution (Carlson *et al.*, 1991; Jørgensen and Rawlings, 2000; Bonn   *et al.*, 2003)

$$\begin{aligned}
\tilde{\phi} &= \sum_{k=0}^{\infty} g_k(y_k, u_k) - g_s(y_s, u_s) \\
&= \frac{1}{2} \sum_{k=0}^{\infty} z'_k Q_y z_k + v'_k R v_k + 2\bar{q}'_k z_k + 2s'_k v_k \\
&= \frac{1}{2} \sum_{k=0}^{\infty} w'_k Q w_k + v'_k R v_k + 2q'_k w_k + 2s'_k v_k
\end{aligned} \tag{1.145}$$

in which we have applied the parameter definitions

$$Q = C' Q_y C \quad q_k = C' \bar{q}_k \tag{1.146}$$

In terms of deviation variables, the general infinite horizon linear quadratic optimal control problem (1.137) may be expressed as

$$\min \quad \phi = \sum_{k=0}^{\infty} \left[ \frac{1}{2} (w'_k Q w_k + v'_k R v_k + 2q'_k w_k + 2s'_k v_k) + g_s(y_s, u_s) \right] \tag{1.147a}$$

$$s.t. \quad w_{k+1} = A w_k + B v_k + E \xi_k \tag{1.147b}$$

$$F w_k + G v_k + H \xi_k \geq 0 \tag{1.147c}$$

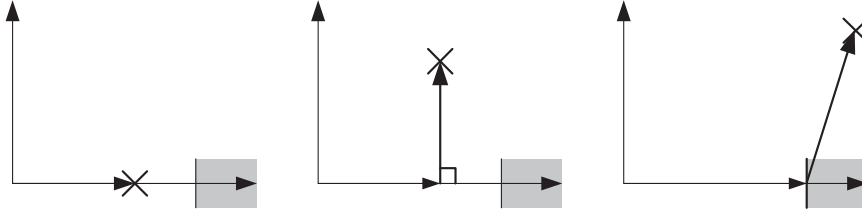
The objective function in this problem may be infinite for all admissible trajectories (J  rgensen and Rawlings, 2000). It is conjectured that the costs are finite if the optimal steady state value are subtracted from the stage costs in the above objective function. This conjecture requires that the optimal solution of the general infinite horizon linear quadratic optimal control problem converges sufficiently fast towards the optimal steady state. The subtraction of a constant, in this case  $g_s(y_s, u_s)$ , at each stage in the objective function does not influence the solution of the dynamic regulation problem. By application of the solution of the steady state problem, the deviation variables are formed such that the regulation problem may be formulated as

$$\min \quad \tilde{\phi} = \frac{1}{2} \sum_{k=0}^{\infty} w'_k Q w_k + v'_k R v_k + 2q'_k w_k + 2s'_k v_k \tag{1.148a}$$

$$s.t. \quad w_{k+1} = A w_k + B v_k + E \xi_k \tag{1.148b}$$

$$F w_k + G v_k + H \xi_k \geq 0 \tag{1.148c}$$

Consider the case in which the output reference is constant,  $r_k = r_s$  for all  $k$ , and the disturbance is constant,  $\xi_k = d_s - d_s = 0$  for all  $k$ . For the case in which no inequality constraints are active at steady state, the linear terms in the objective function of (1.148) vanishes at the optimal solution. As illustrated in figure 1.2, the linear terms may either vanish because  $y_s = r_s$  and



**Figure 1.2.** Illustration of the 3 different principle situations determining the existence of linear terms in the infinite horizon general linear quadratic optimal control problem. The first axis until the gray shaded area represents the admissible space. The crosses represents set-points which may be unreachable. In the left figure, the setpoint is reachable and the linear terms vanish as target value equals the reached value at steady state. In the middle figure, the set point is unreachable. However, at the optimal solution no constraints are active at steady state and therefore the setpoint error vector is orthogonal (conjugate) to the optimal solution. This orthogonality (conjugacy) implies that the linear terms do not have any practical effect on the solution as they will be zero at the optimal solution. In the third case, illustrated by the right figure, the set point is not reachable and a constraint is active at steady state. This implies that the linear terms in the objective function will be non-zero at the optimal solution and have an effect on the optimal solution.

$u_s = \bar{u}$  such that the coefficients of the linear terms are zero, or due to the orthogonality property of optimal solutions in Hilbert spaces (Luenberger, 1969). The tracking error vector will only be orthogonal to the optimal solution if no constraints are strictly active at steady state. In these situations at which the linear terms in the objective function vanish, the dynamic regulation problem is equal to the standard infinite horizon constrained regulation problem (1.148)

$$\min \quad \tilde{\phi} = \frac{1}{2} \sum_{k=0}^{\infty} w_k' Q w_k + v_k' R v_k \quad (1.149a)$$

$$s.t. \quad w_{k+1} = A w_k + B v_k \quad (1.149b)$$

$$F w_k + G v_k \geq 0 \quad (1.149c)$$

and may be solved employing the techniques for solution of infinite horizon constrained problems for which origo is in the interior of the feasible region.

For the case when constraints are active at steady state, the linear terms in the objective function of (1.148) do not vanish and it is important for the qualitative properties of the solution that they are retained in the formulation of the quadratic program (Jørgensen and Rawlings, 2000). Due to the linear terms and active steady state constraints implying that origo is on the boundary of the feasible region, (1.148) can be solved approximately by the techniques of Rao

and Rawlings (1999) and Pannocchia *et al.* (2002). With some modifications these techniques are applicable to the problem with linear terms in the objective function, even though neither Rao and Rawlings (1999) nor Pannocchia *et al.* (2002) included linear terms in their solution procedures for infinite horizon constrained linear quadratic regulation problems. Active constraints at steady state may for instance arise in situations at which a large disturbance makes it impossible to keep the process at the specified set point.

### 1.2.4 Infinite-Horizon Optimal Control

The procedures proposed in the literature for solution of the infinite-horizon optimal control problem are developed mainly for the regulation problem and require construction of the maximum output admissible set. In contrast to the pure regulation problem, the maximum output admissible set cannot be constructed off-line for the general optimal control problem but must be constructed on-line for the actual disturbances estimated. This construction may be computational expensive. An alternative approach may be to solve a *finite* horizon optimal control problem with a sufficient long horizon and rely on convergence of the finite horizon program towards the infinite horizon program. For some systems having the *turnpike* property this convergence may be detected by monitoring whether the solution stays close to the optimal steady state for an extended period of time. This section describes the turnpike property concept as well as optimality definitions for general infinite horizon optimal control problems.

For stability reasons, the model predictive control problem has been formulated as infinite-horizon optimal control problems (Mayne *et al.*, 2000)

$$\min \quad \phi = \sum_{k=0}^{\infty} g(x_k, u_k) \quad (1.150a)$$

$$s.t. \quad x_{k+1} = f(x_k, u_k) \quad (1.150b)$$

$$x_0 = a \quad (1.150c)$$

$$u_k \in \mathcal{U}(x_k) \quad (1.150d)$$

The admissible state-control trajectories of this infinite horizon optimal control are denoted by the set

$$\mathcal{A}_{\infty} = \{\{x_k, u_k\}_{k=0}^{\infty} : x_0 = a, x_{k+1} = f(x_k, u_k), u_k \in \mathcal{U}(x_k)\}$$

In the formulation of the infinite-horizon optimal control problem constituting the model predictive controller, it has commonly been either tacitly or explicitly assumed that there exist admissible state-control sequences such that the objective function

$$\phi = \sum_{k=0}^{\infty} g(x_k, u_k) \quad \{x_k, u_k\}_{k=0}^{\infty} \in \mathcal{A}_{\infty}$$

is finite or that an equivalent regulation problem with finite cost may be constructed. In the Muske-Rawlings approach (Muske and Rawlings, 1993b), this regulation problem is constructed by computation of the solution  $(x_{ss}^*, u_{ss}^*)$  of the associated steady state problem

$$\min_{x,u} g(x, u) \quad (1.151a)$$

$$s.t. \quad x = f(x, u) \quad (1.151b)$$

$$u \in \mathcal{U}(x) \quad (1.151c)$$

and construction of the modified infinite-horizon optimal control problem

$$\min \quad \tilde{\phi} = \sum_{k=0}^{\infty} \tilde{g}(x_k, u_k) = \sum_{k=0}^{\infty} [g(x_k, u_k) - g(x_{ss}^*, u_{ss}^*)] \quad (1.152a)$$

$$s.t. \quad x_{k+1} = f(x_k, u_k) \quad (1.152b)$$

$$x_0 = a \quad (1.152c)$$

$$u_k \in \mathcal{U}(x_k) \quad (1.152d)$$

The motivation for this construction is *ad hoc* and seems to rely on the conjecture that the optimal state-control trajectory  $\{x_k, u_k\}_{k=0}^{\infty}$  converges sufficiently fast to the optimal steady state  $(x_{ss}^*, u_{ss}^*)$ . Sufficiently fast convergence means that the convergence of the stage cost  $g(x_k^*, u_k^*)$  towards  $g(x_{ss}^*, u_{ss}^*)$  is such that the objective function is finite

$$\tilde{\phi}^* = \sum_{k=0}^{\infty} \tilde{g}(x_k^*, u_k^*) = \sum_{k=0}^{\infty} [g(x_k^*, u_k^*) - g(x_{ss}^*, u_{ss}^*)] < \infty$$

Consider the stabilizable and detectable linear-quadratic control problem with constraints. If no constraints are strictly active at steady state, the costs are finite as this is essentially a linear-quadratic *regulation* problem after some finite transition period at which the inequalities may be active. In this situation, the obtained modified infinite-horizon optimal control problem is similar to (1.149). In the case, when inequality constraints are strictly active at steady state the obtained infinite-horizon optimal control problem is of the type (1.148)<sup>2</sup>.

While the stability properties of the infinite horizon regulation problem are well-studied by the model predictive control community (Mayne *et al.*, 2000), the properties and concepts of the general infinite-horizon optimal control problem remain to be explained in relation to model predictive control. Carlson and Haurie (1987) and Carlson *et al.* (1991) provide a comprehensive treatment of the continuous-time infinite-horizon optimal control problem. Weizsäcker (1965), Halkin (1974), Brock and Haurie (1976), Haurie (1976, 1980), Feinstein and Luenberger (1981), Stern (1984), Takayama (1985), Artstein and

<sup>2</sup>It remains to be established that the optimal objective function of this program is finite. However, under fairly mild conditions the optimal state-control trajectory converges to the optimal steady state (Bonné *et al.*, 2003).



Leizarowitz (1985), Leizarowitz (1987a,b), Carlson (1986a,b,c, 1987, 1990), Carlson *et al.* (1987), Barbieri and Alba-Flores (2000), and Blot and Hayek (2001) describe issues related to general infinite-horizon optimal control problems.

One issue that need to be addressed in the exploration of general infinite horizon optimal control problems is the definition of optimality (Weizsäcker, 1965; Hammond and Mirrless, 1973; Hammond and Kennan, 1979; Haurie and Sethi, 1984; Stern, 1984; Carlson, 1990). It is not possible in general to extend the optimality definition of finite dimensional nonlinear programs to infinite horizon optimal control problems as the problem may be well defined even though the cost is infinite for all admissible state-control trajectories (Jørgensen and Rawlings, 2000). This observation calls for weaker optimality notions than employed when a finite optimal cost exists.

**Definition 1.2.2 (Infinite Horizon Optimal Control Optimality Concepts)**

Let  $(x, u) = \{x_k, u_k\}_{k=0}^{\infty}$ , let the admissible set be

$$\mathcal{A}_{\infty} = \{\{x_k, u_k\}_{k=0}^{\infty} : x_0 = a, x_{k+1} = f(x_k, u_k), u_k \in \mathcal{U}(x_k)\} \quad (1.153)$$

and consider the mathematical program

$$\phi_N(x^*, u^*) = \min_{(x, u)} \left\{ \phi_N(x, u) = \sum_{k=0}^{N-1} g(x_k, u_k) : (x, u) \in \mathcal{A}_{\infty} \right\} \quad (1.154)$$

which is identical to (1.150) except that it has a finite dimensional objective function.

An admissible pair  $(x^*, u^*) \in \mathcal{A}_{\infty}$  is called

1. **Strongly optimal** of (1.150) if

$$\phi_{\infty}(x^*, u^*) = \lim_{N \rightarrow \infty} \phi_N(x^*, u^*) \text{ is finite} \quad (1.155a)$$

$$\forall (x, u) \in \mathcal{A}_{\infty} \exists \hat{N}(x, u) \geq 0 : \forall N \geq \hat{N}(x, u) : \phi_N(x^*, u^*) \leq \phi_N(x, u) \quad (1.155b)$$

2. **Overtaking optimal** (catching up optimal) of (1.150) if, for all  $\varepsilon > 0$  and  $(x, u) \in \mathcal{A}_{\infty}$  there exists a  $\hat{N} = \hat{N}(\varepsilon, x, u) \geq 0$  such that for all  $N \geq \hat{N}$ :

$$\phi_N(x^*, u^*) \leq \phi_N(x, u) + \varepsilon \quad (1.156)$$

3. **Weakly overtaking optimal** (sporadically catching up optimal) of (1.150) if, for all  $\varepsilon > 0$ ,  $(x, u) \in \mathcal{A}_{\infty}$ , and  $\hat{N} \geq 0$ , there exists a  $N(\varepsilon, x, u) \geq \hat{N}$  such that

$$\phi_N(x^*, u^*) \leq \phi_N(x, u) + \varepsilon \quad (1.157)$$

Overtaking optimality and weakly overtaking optimality do not require the optimal value of the objective function to be finite, while strong optimality does. Strong optimality is equivalent to the usual finite dimensional optimality notion and may be denoted

$$\phi_{\infty}(x, u) \geq \phi_{\infty}(x^*, u^*) \quad \forall (x, u) \in \mathcal{A}_{\infty} \quad (1.158)$$

which is equivalent to

$$\lim_{N \rightarrow \infty} \phi_N(x, u) - \lim_{N \rightarrow \infty} \phi_N(x^*, u^*) \geq 0 \quad \forall (x, u) \in \mathcal{A}_\infty \quad (1.159)$$

under the assumption that  $\phi_\infty(x^*, u^*) = \lim_{N \rightarrow \infty} \phi_N(x^*, u^*)$  is finite. For comparison, the notion of overtaking optimality may be expressed as

$$\lim_{N \rightarrow \infty} \left[ \inf_{n \geq N} \{ \phi_n(x, u) - \phi_n(x^*, u^*) \} \right] \geq 0 \quad \forall (x, u) \in \mathcal{A}_\infty \quad (1.160)$$

Similarly, the notion of weakly overtaking optimality is equivalent to

$$\lim_{N \rightarrow \infty} \left[ \sup_{n \geq N} \{ \phi_n(x, u) - \phi_n(x^*, u^*) \} \right] \geq 0 \quad \forall (x, u) \in \mathcal{A}_\infty \quad (1.161)$$

The *turnpike property* for some infinite horizon optimal control problems was first discovered in the context of optimal growth models by Samuelson (1966). A system is said to have the turnpike property if the optimal state-control trajectory of an optimal control problem with either a finite or an infinite horizon stays close to the corresponding optimal steady state for much of the time. Hence, the optimal state-control trajectory of a finite horizon optimal control problem has the turnpike property if it after some initial transient approaches the optimal steady state and only diverges from this steady state close to the terminal time. An infinite horizon optimal control problem has the turnpike property if the optimal state-control trajectory converges asymptotically towards the optimal steady state-control pair. The turnpike property is desirable, as it indicates that the initial part of the solution of the infinite-horizon optimal control problem may be obtained at arbitrary precision by solving a finite horizon optimal control problem and verifying that the state-control trajectory remains in the vicinity of the corresponding optimal steady state for some time. If the infinite-horizon optimal control problem has the turnpike property, then the corresponding finite-horizon optimal control problem will also have the turnpike property for a sufficient long horizon.

Bonné *et al.* (2003) apply a methodology proposed by Brock and Haurie (1976) to convex discrete-time infinite-horizon optimal control problems. Preliminary results suggest that an overtaking minimizer of a convex infinite-horizon optimal control problems has the turnpike property. The constrained linear quadratic regulation problem with possible steady state active constraints is a convex infinite-horizon optimal control problem which may have unbounded optimal cost. Hence, this problem may most likely be efficiently solved in general by utilization of the turnpike property. The exact strategy for exploitation of the turnpike property remains an open question.

## 1.3 Constraints

The industrial success of model predictive control has to a large extent been attributed to its ability to incorporate constraints in a rigorous way. This an important ability for control of industrial processes as the optimal point of operation tends to be in the vicinity of constraints. Model predictive control is the only control methodology that is able to cope with constraints in a systematic manner. There are two major types of constraints in model predictive control: Hard constraints and soft constraints.

### 1.3.1 Hard Constraints

One class of hard constraints is constraints that can never be violated due to physical limitations of the equipment. These constraints are typically actuator constraints also called input constraints. In example, the out flow from a pump must be between a lower limit (say zero) and the maximum capacity of the pump, the valve opening must be between zero and one, and so on. In the mathematical program constituting the model predictive controller, they are represented by bounds of the type

$$u_{\min} \leq u_k \leq u_{\max} \quad (1.162)$$

Quite often, actuator equipment is not only limited by the range in which it can operate but also by the rate at which it can change value within this range. Neither a pump nor a valve can instantaneously change from its lower limit to its upper limit. Such rate-of-change limitations on the actuator equipment is modeled by inequalities of the type

$$-\Delta \leq \Delta u_k \leq \Delta \quad (1.163)$$

in which  $\Delta u_k = u_k - u_{k-1}$ . The limit,  $\Delta$ , on the rate of change may also be applied as a tuning parameter preventing the controller from making abrupt changes in the input variables. If the rate of change is physically motivated and the maximum rate of change in continuous time is given by the vector  $\delta$ , then the rate of change in the discrete time setting is obtained as:  $\Delta = T_s \delta$  in which  $T_s$  is the sampling period.

The main characteristic feature of input constraints and input rate-of-change constraints is that they are always feasible. This implies that the mathematical

program

$$\min \quad \phi = \frac{1}{2} \sum_{k=0}^{\infty} (y_k - r)' Q (y_k - r) + u_k' R u_k + \Delta u_k' \Phi \Delta u_k \quad (1.164a)$$

$$s.t. \quad x_{k+1} = A x_k + B u_k \quad (1.164b)$$

$$y_k = C x_k \quad (1.164c)$$

$$u_{\min} \leq u_k \leq u_{\max} \quad (1.164d)$$

$$-\Delta \leq \Delta u_k \leq \Delta \quad (1.164e)$$

constituting the model predictive controller with hard input constraints is guaranteed to be feasible. The quadratic program (1.164) will therefore always have a solution. This property is compatible with the physically motivated fact that input constraints must be satisfied.

Output constraints and general state constraints associated with the dynamic system

$$x_{k+1} = A x_k + B u_k \quad (1.165a)$$

$$y_k = C x_k \quad (1.165b)$$

are of a different nature and may be infeasible for all admissible control sequences  $\{u_k\}$ . Output constraints and state constraints may be used to represent desirable operating regions, safety constraints, and constraints imposed by environmental regulations. As an example consider a storage tank for a liquid. The volume of liquid in the tank must be between an upper and a lower limit. The absolute lower limit is zero. Physics dictates that this limit never will be violated. An absolute upper limit for the volume of liquid in the tank is the capacity of the tank. However, this constraint may be violated with the result of the tank flowing over. Therefore, it is desirable to impose the constraint that the volume of liquid in the tank should be less than its capacity. This is an example of an output constraint motivated by a desirable operating regime.

Plain output constraints of this system are modeled as

$$y_{\min} \leq y_k \leq y_{\max} \quad (1.166)$$

while general state constraints are expressed by the inequality

$$G x_k + H u_k \geq b \quad (1.167)$$

If the output and state constraints are modeled as hard constraints, the mathe-

mathematical program constituting the model predictive controller becomes

$$\min \quad \phi = \frac{1}{2} \sum_{k=0}^{\infty} (y_k - r)' Q (y_k - r) + u_k' R u_k + \Delta u_k' \Phi \Delta u_k \quad (1.168a)$$

$$s.t. \quad x_{k+1} = A x_k + B u_k \quad (1.168b)$$

$$y_k = C x_k \quad (1.168c)$$

$$u_{\min} \leq u_k \leq u_{\max} \quad (1.168d)$$

$$-\Delta \leq \Delta u_k \leq \Delta \quad (1.168e)$$

$$y_{\min} \leq y_k \leq y_{\max} \quad (1.168f)$$

$$G x_k + H u_k \geq b \quad (1.168g)$$

The disadvantage of having hard output and state constraints in the model predictive controller is that the mathematical program (1.168) may become infeasible. Hence, model predictive controllers with output and state constraints have to have some mechanism to recover from situations with infeasible output and state constraints. Of course, one element of such a recovery procedure could be to throw an alarm. In safety critical situations it could even be to start a plant shut-down procedure. Another element of the recovery procedure could and often is to reformulate the mathematical program such that the model predictive controller takes the in some sense best possible action in face of not being able to meet all output constraints.

### 1.3.2 Soft Constraints Motivated by Infeasibilities

The situation with infeasible output constraints has been addressed by essentially three different approaches. The first approach is the minimum time approach in which the duration of the infeasible output constraints is minimized (Rawlings and Muske, 1993; Muske and Rawlings, 1993b; Scokaert and Rawlings, 1999). The second approach is the soft constraint approach in which some measure of the size of the output constraint violations is minimized (Ricker *et al.*, 1988; de Oliveira and Biegler, 1994; Zheng and Morari, 1995; Scokaert and Rawlings, 1999; Rao and Rawlings, 1999; Rawlings, 1999, 2000; Kerrigan and Maciejowski, 2000; Pannocchia *et al.*, 2001, 2002). The third major class of approaches is the prioritized output constraint approach (Tyler and Morari, 1999; Vada *et al.*, 1999; Vada, 2000; Strand, 2003). In this method, the output constraints are prioritized according to their importance. If an infeasible situation occur, the output constraints are relaxed in succession according to their importance using the soft constraint approach until a feasible mathematical program is constructed.

For convenience, we will denote output and state constraints as

$$C x_k + D u_k \geq b \quad (1.169)$$

in the following discussion.

Rawlings and Muske (1993) demonstrated that there exists a minimum time,  $\kappa(x_0)$ , dependent on the initial state,  $x_0$ , such that the output constraints in the mathematical program

$$\min_{\{x_k, u_k\}} \quad \phi = \frac{1}{2} \sum_{k=0}^{\infty} x_k' Q x_k + u_k' R u_k \quad (1.170a)$$

$$s.t. \quad x_{k+1} = A x_k + B u_k \quad (1.170b)$$

$$C x_k + D u_k \geq b \quad k = \kappa(x_0), \kappa(x_0) + 1, \dots \quad (1.170c)$$

$$u_k = K x_k \quad k = N, N + 1, \dots \quad (1.170d)$$

constituting the model predictive controller will be feasible on an infinite horizon. This is the minimum-time approach to recover from infeasibilities. In this predictive controller, the output constraints are discarded until time  $\kappa(x_0)$ . Rawlings and Muske (1993) provide a method to compute an upper bound on  $\kappa(x_0)$ . An alternative method for computing an upper bound for  $\kappa(x_0)$  is to detect the times for which the MPC without output constraints violates them. Starting from this bound  $\kappa(x_0)$  is decreased until the following LP becomes infeasible

$$\min \quad \psi = s' \varepsilon \quad (1.171a)$$

$$x_{k+1} = A x_k + B u_k \quad (1.171b)$$

$$C x_k + D u_k \geq b - \varepsilon \quad k = 0, 1, \dots, \kappa(x_0) - 1 \quad (1.171c)$$

$$C x_k + D u_k \geq b \quad k = \kappa(x_0), \kappa(x_0) + 1, \dots \quad (1.171d)$$

$$u_k = K x_k \quad k = N, N + 1, \dots \quad (1.171e)$$

The value of  $\kappa(x_0)$  computed in this way is subsequently used in the construction and solution of (1.170). If the optimal value of (1.171) for any proposed value of  $\kappa(x_0)$  becomes zero, then  $\kappa(x_0) = 0$  and the output constraints are feasible. The slack variables  $\varepsilon$  represent the maximum constraint violation but are not used directly in (1.170). Scokaert and Rawlings (1999) proposed to use the optimal value of the slack variables,  $\varepsilon^*$ , such that the size of the minimal duration output constraint violation is minimized. This predictive controller is called the optimized minimal-time MPC and its quadratic program is

$$\min \quad \phi = \frac{1}{2} \sum_{k=0}^{\infty} x_k' Q x_k + u_k' R u_k \quad (1.172a)$$

$$s.t. \quad x_{k+1} = A x_k + B u_k \quad (1.172b)$$

$$C x_k + D u_k \geq b - \varepsilon^* \quad k = 0, 1, \dots, \kappa(x_0) - 1 \quad (1.172c)$$

$$C x_k + D u_k \geq b \quad k = \kappa(x_0), \kappa(x_0) + 1, \dots \quad (1.172d)$$

$$u_k = K x_k \quad k = N, N + 1, \dots \quad (1.172e)$$

which is always feasible due to the construction of  $\varepsilon^*$ . The pros and cons of this constraint relaxation procedure has been discussed by Scokaert and Rawlings (1999).

The soft constraint approach is another methodology to relax output constraints and avoid infeasible mathematical programs. Ricker *et al.* (1988) as well as Zheng and Morari (1995) relaxed the output constraints by introduction of slack variables,  $\varepsilon$ , and penalized the maximal violation of the output constraints as indicated in the following mathematical program

$$\min_{\{x_k, u_k\}, \varepsilon} \quad \phi = \frac{1}{2} \left( \sum_{k=0}^{\infty} x'_k Q x_k + u'_k R u_k \right) + \frac{1}{2} \varepsilon' S \varepsilon \quad (1.173a)$$

$$s.t. \quad x_{k+1} = A x_k + B u_k \quad k = 0, 1, \dots \quad (1.173b)$$

$$C x_k + D u_k \geq b - \varepsilon \quad k = 0, 1, \dots \quad (1.173c)$$

$$\varepsilon \geq 0 \quad (1.173d)$$

The output constraint relaxation procedure allows violation of the output constraints but penalizes them to avoid such violations if possible. However, the output constraints may be violated in this formulation even if that could be avoided. In addition to this disadvantage, penalizing the peak violation introduces a mismatch between the open-loop predictions and the actual closed-loop behavior (Scokaert and Rawlings, 1999). This leads to counter-intuitive behavior and degrades performance.

Violation of the output constraints due to the softening when they are feasible can be avoided by application of *exact* soft constraints (Fletcher, 1987; de Oliveira and Biegler, 1994; Kerrigan and Maciejowski, 2000). In this approach, a linear term,  $s'\varepsilon$ , in the slack variables,  $\varepsilon$ , is added to the objective function

$$\min_{\{x_k, u_k\}, \varepsilon} \quad \phi = \frac{1}{2} \left( \sum_{k=0}^{\infty} x'_k Q x_k + u'_k R u_k \right) + \frac{1}{2} \varepsilon' S \varepsilon + s' \varepsilon \quad (1.174a)$$

$$s.t. \quad x_{k+1} = A x_k + B u_k \quad k = 0, 1, \dots \quad (1.174b)$$

$$C x_k + D u_k \geq b - \varepsilon \quad k = 0, 1, \dots \quad (1.174c)$$

$$\varepsilon \geq 0 \quad (1.174d)$$

If the coefficients,  $s > 0$ , are selected sufficiently large (larger than the maximal Lagrange multipliers) the linear term guarantees that the output constraints are not violated if that can be avoided. In practice,  $S$  as well as  $s$  should be regarded as tuning knobs that should be chosen large enough to efficiently penalize any output constraint violation.

The mismatch between the open-loop predictions and the closed-loop behavior observed for soft constraints penalizing the peak violation can be avoided by penalizing the sum of violations instead (Scokaert and Rawlings, 1999). In this case, the mathematical program constituting the model predictive controller

may be expressed as

$$\min_{\{x_k, u_k, \varepsilon_k\}} \quad \phi = \sum_{k=0}^{\infty} \frac{1}{2} (x_k' Q x_k + u_k' R u_k + \varepsilon_k' S \varepsilon_k) + s' \varepsilon_k \quad (1.175a)$$

$$s.t. \quad x_{k+1} = A x_k + B u_k \quad k = 0, 1, \dots \quad (1.175b)$$

$$C x_k + D u_k \geq b - \varepsilon_k \quad k = 0, 1, \dots \quad (1.175c)$$

$$\varepsilon_k \geq 0 \quad k = 0, 1, \dots \quad (1.175d)$$

By introducing the variables

$$v_k = \begin{bmatrix} u_k \\ \varepsilon_k \end{bmatrix}$$

$$\bar{B} = \begin{bmatrix} B & 0 \end{bmatrix} \quad \bar{C} = \begin{bmatrix} C \\ 0 \end{bmatrix} \quad \bar{D} = \begin{bmatrix} D & I \\ 0 & I \end{bmatrix}$$

$$\bar{R} = \begin{bmatrix} R & 0 \\ 0 & S \end{bmatrix} \quad \bar{s} = \begin{bmatrix} 0 \\ s \end{bmatrix} \quad q = 0$$

(1.175) may be reformulated as a standard constrained linear quadratic regulation problem

$$\min_{\{x_k, v_k\}} \quad \phi = \sum_{k=0}^{\infty} \frac{1}{2} (x_k' Q x_k + v_k' \bar{R} v_k) + q' x_k + \bar{s}' v_k \quad (1.176a)$$

$$s.t. \quad x_{k+1} = A x_k + \bar{B} v_k \quad k = 0, 1, \dots \quad (1.176b)$$

$$\bar{C} x_k + \bar{D} v_k \geq \bar{b} \quad k = 0, 1, \dots \quad (1.176c)$$

which due its construction is guaranteed to be feasible. First of all, note that (1.176) illustrates the need for linear terms in the objective function even when the basic problem has its dynamics and objective function described in the standard LQR fashion. Secondly, note that the introduction of slack variables  $\{\varepsilon_k\}$  increases the size of the mathematical program (1.175) considerably. However, the special optimal control problem structure can be efficiently exploited in the structure utilizing solution procedures.

Next we consider the formulation of a practical model predictive controller with soft constraints. For setpoint tracking model predictive controllers, the objective function is often expressed as

$$\phi = \sum_{k=0}^{\infty} \frac{1}{2} ((y_k - r)' Q_y (y_k - r) + u_k' R u_k + \Delta u_k' \Phi \Delta u_k + \varepsilon_k' S \varepsilon_k) + s' \varepsilon_k \quad (1.177)$$

For this case with pure output constraints, we may formulate the mathematical



program constituting the model predictive controller with soft constraints as

$$\min_{\{x_k, u_k, \varepsilon_k\}} \phi \quad (1.178a)$$

$$s.t. \quad x_{k+1} = Ax_k + Bu_k \quad (1.178b)$$

$$y_k = Cx_k \quad (1.178c)$$

$$u_{\min} \leq u_k \leq u_{\max} \quad (1.178d)$$

$$-\Delta_u \leq \Delta u_k \leq \Delta_u \quad (1.178e)$$

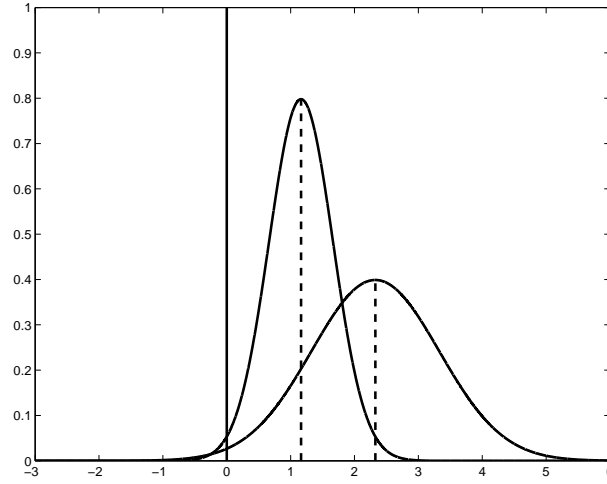
$$y_{\min} - \varepsilon_k \leq y_k \leq y_{\max} + \varepsilon_k \quad (1.178f)$$

$$\varepsilon_k \geq 0 \quad (1.178g)$$

## 1.4 The Control Problem

In the process industries, the control problem for continuous processes occurs in two main forms. One control problem concerns the switch of product or quality produced. In this case, the objective is usually to minimize the transition time such that off-spec product produced is minimized and such that more product is produced. Application of nonlinear model predictive control is one promising methodology for minimizing this transition time between recipes. The other major control problem in the process industries concerns keeping the process at a set point. This set point may be the result of a steady state optimization. The optimal operating point at which the process is operated is often located in the vicinity of output constraints. Due to random disturbances entering the process, these constraints would be frequently violated if the set point specified is at the constraints or very close them. Therefore, depending on the variance of the output variables, it is necessary to back-off from the constraints such that they are respected with a certain probability, say 99%. As explained by Richalet *et al.* (1978), the economic value of advanced regulation is not the variance reduction in itself, but rather the reduced production costs stemming from the ability of being able to move closer to some constraint due to the variance reduction. This effect is illustrated in figure 1.3.

The dynamic economic effect of applying model predictive control in conjunction with real time optimization has been studied by Loeblein and Perkins (1999a,b). They applied the economic dynamic concept introduced by Narraway *et al.* (1991) and also explained by Heath (1996). In their approach, marginal distributions of the output variables in an unconstrained model predictive controller is computed. These marginal distributions are used in computing the *approximate* back-off such that the output constraints are satisfied with a given probability. Kassmann *et al.* (2000) applied some of the same concepts in robust computation of the steady-state target for model predictive control. Loeblein and Perkins (1999a) computed the back-off for an unconstrained Muske-Rawlings model predictive controller. A similar procedure may be applied to the LQG controller which is the unconstrained version of



**Figure 1.3.** The necessary back-off from a lower bound on the output. The plot shows the probability distribution of two normally distributed variables with different variance and mean. The distributions are constructed such the corresponding stochastic variables each satisfy the constraint  $x \geq 0$  with a probability of 99%. The variable with greater variance need a larger back-off from the lower limit which translates into a higher production cost. One motive of advanced control is to reduce this variance such that the back-off is reduced and with lower production cost as result.

the constrained linear quadratic controller. In this procedure it can be utilized that the involved distributions are Gaussian. The variance of the closed loop system may be computed and analyzed using an eigenvalue decomposition of the covariance matrix (Poulsen, 1995a, chap. 3).

The next two examples show the variance reducing effect of one type of LQG control which is one type of model predictive control.

#### Example 1.4.1 (Closed Loop Properties of a LQG System)

In this example the above procedure is illustrated for a LQG system. The derivations and principles employed are adopted from Söderström (2002) and Poulsen (1995b).

Consider the system

$$x_{k+1} = Ax_k + Bu_k + Gw_k \quad (1.179a)$$

$$y_k = Cx_k + v_k \quad (1.179b)$$

$$z_k = Dx_k \quad (1.179c)$$

in which

$$\begin{bmatrix} w_k \\ v_k \end{bmatrix} \in N_{iid} \left( \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} Q & S \\ S' & R \end{bmatrix} \right) \quad x_0 \in N(\hat{x}_{0|-1}, P_{0|-1}) \quad (1.180)$$

$x_k$  is the states,  $u_k$  the controls,  $y_k$  the measurements, and  $z_k$  the outputs to be controlled.

Provided the system is stable the asymptotic open loop covariance,  $P_{ol}$ , of the state vector may be computed as the solution of the discrete Lyapunov equation

$$P_{ol} = AP_{ol}A' + GQG \quad (1.181)$$

From the covariance of the states, the measurement covariance,  $\Lambda_{ol}$ , and the output covariance,  $\Gamma_{ol}$ , may be computed by the expressions

$$\Lambda_{ol} = CP_{ol}C' + R \quad (1.182a)$$

$$\Gamma_{ol} = DP_{ol}D' \quad (1.182b)$$

The open loop is characterized by  $u_k = 0$ . Then for stable systems in the limit  $k \rightarrow \infty$  the mean evolution of the system may be expressed as

$$\hat{x}_{k+1} = A\hat{x}_k \rightarrow 0 \quad (1.183a)$$

$$\hat{y}_k = C\hat{x}_k \rightarrow 0 \quad (1.183b)$$

$$\hat{z}_k = D\hat{x}_k \rightarrow 0 \quad (1.183c)$$

and the distribution of the states, the measurements and the outputs are normal. Hence, in the open loop case

$$k \rightarrow \infty : \quad x_k \in N(0, P_{ol}) \quad y_k \in N(0, \Lambda_{ol}) \quad z_k \in N(0, \Gamma_{ol})$$

The LQG controller of (1.179) is the solution the stochastic optimization problem which may loosely be stated as

$$\min_{\{x_k, u_k, z_k\}_{k=0}^{\infty}} E \left\{ \frac{1}{2} \sum_{k=0}^{\infty} z_k' Q z_k \right\} \quad (1.184a)$$

$$s.t. \quad x_{k+1} = Ax_k + Bu_k + Gw_k \quad (1.184b)$$

$$y_k = Cx_k + v_k \quad (1.184c)$$

$$z_k = Dx_k \quad (1.184d)$$

for which it is understood that the measurements arrive sequentially and the optimal control is causal. Under mild conditions such as stabilizability and detectability, the solution to this problem may be derived analytically and given by separation into a controller and an estimator.

The controller is given by

$$u_k = L\hat{x}_{k|k} \quad (1.185)$$

in which  $\hat{x}_{k|k} = E\{x_k | \mathcal{I}_k\}$  is the conditional mean with the information vector given as  $\mathcal{I}_k = \{y_0, u_0, \dots, y_{k-1}, u_{k-1}, y_k\}$ . The gain  $L$  is computed as part in the solution of a Riccati equation

$$P_{lq} = A'P_{lq}A + D'Q_zD - A'P_{lq}B(B'P_{lq}B)^{-1}B'P_{lq}A \quad (1.186a)$$

$$L = -(B'P_{lq}B)^{-1}B'P_{lq}A \quad (1.186b)$$

The estimator, considered in the limit  $k \rightarrow \infty$  is the static Kalman filter

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_f(y_k - C\hat{x}_{k|k-1}) \quad (1.187a)$$

$$\hat{w}_{k|k} = K_w(y_k - C\hat{x}_{k|k-1}) \quad (1.187b)$$

$$\hat{x}_{k+1|k} = A\hat{x}_{k|k} + Bu_k + G\hat{w}_{k|k} \quad (1.187c)$$

with the gains computed by the expressions

$$P_p = AP_pA' + GQG' - (AP_pC' + GS)(CP_pC' + R)^{-1}(AP_pC' + GS)' \quad (1.188a)$$

$$R_e = CP_pC' + R \quad (1.188b)$$

$$K_f = P_pC'R_e^{-1} \quad (1.188c)$$

$$K_w = SR_e^{-1} \quad (1.188d)$$

$P_p$  is obtained as solution of the stated discrete-time Riccati equation. In the limit  $k \rightarrow \infty$ , the state estimate conditioned on the information  $\mathcal{I}_{k|k}$  is normally distributed with mean  $\hat{x}_{k|k}$  and covariance

$$P_f = P_p - K_fR_e^{-1}K_f' \quad (1.189)$$

Hence

$$k \rightarrow \infty : \quad x_k | \mathcal{I}_k \in N(\hat{x}_{k|k}, P_f) \quad (1.190)$$

Furthermore

$$k \rightarrow \infty : \quad x_k | \mathcal{I}_{k-1} \in N(\hat{x}_{k|k-1}, P_p) \quad (1.191)$$

Define the predictive state estimation error,  $\tilde{x}_{k|k-1}$ , as

$$\tilde{x}_{k|k-1} = x_k - \hat{x}_{k|k-1} \quad (1.192)$$

Then  $\tilde{x}_{k|k-1}$  is normally distributed (Poulsen, 1995b)

$$k \rightarrow \infty : \quad \tilde{x}_{k|k-1} \in N(0, P_p) \quad (1.193)$$

and  $\tilde{x}_{k|k-1}$  is independent of  $\hat{x}_{k|k-1}$ . This independence is easily memorized using the inner product approach described by Kailath *et al.* (2000) and the fact that the error vector,  $\tilde{x}_{k|k-1}$ , and the optimal estimate,  $\hat{x}_{k|k-1}$ , are orthogonal. The orthogonality implies that  $\tilde{x}_{k|k-1}$  and  $\hat{x}_{k|k-1}$  are uncorrelated. Normally distributed vectors that are uncorrelated are also independent.

Using the system dynamics, the dynamics for the estimator, the controller equation, and the definition of the estimation error,  $\tilde{x}_{k|k-1}$ , it is possible to derive the following expression

$$\begin{bmatrix} x_{k+1} \\ \tilde{x}_{k+1|k} \end{bmatrix} = \begin{bmatrix} A + BL & BL(K_fC - I) \\ 0 & A - (AK_f + GK_w)C \end{bmatrix} \begin{bmatrix} x_k \\ \tilde{x}_{k|k-1} \end{bmatrix} + \begin{bmatrix} G & BLK_f \\ G & -(AK_f + GK_w) \end{bmatrix} \begin{bmatrix} w_k \\ v_k \end{bmatrix} \quad (1.194)$$

for the evolution of the closed loop system. This system may be expressed as

$$\bar{x}_{k+1} = \bar{A}\bar{x}_k + \bar{G}\bar{w}_k \quad (1.195)$$

in which the variables  $\bar{x}_k$  and  $\bar{w}_k$  are defined as

$$\bar{x}_k = \begin{bmatrix} x_k \\ \tilde{x}_{k|k-1} \end{bmatrix} \quad \bar{w}_k = \begin{bmatrix} w_k \\ v_k \end{bmatrix} \quad (1.196)$$

The matrices  $\bar{A}$  and  $\bar{G}$  are defined as

$$\bar{A} = \begin{bmatrix} A + BL & BL(K_f C - I) \\ 0 & A - (AK_f + GK_w)C \end{bmatrix} \quad \bar{G} = \begin{bmatrix} G & BLK_f \\ G & -(AK_f + GK_w) \end{bmatrix} \quad (1.197)$$

Then the stationary variance of the vector  $\bar{x}_k$  may be computed as the solution of the Lyapunov equation

$$\bar{P} = \bar{A}\bar{P}\bar{A}' + \bar{G}\bar{Q}\bar{G}' \quad (1.198)$$

in which the matrix  $\bar{P}$  is partitioned as

$$\bar{P} = \begin{bmatrix} P_{cl} & P_{12} \\ P_{21} & P_p \end{bmatrix} \quad (1.199)$$

$P_{cl}$  is the stationary covariance of the state vector  $x_k$  obtained in closed loop with the estimator and regulator. The corresponding covariances of the measurement,  $y_k$ , and the output,  $z_k$ , are

$$\Lambda_{cl} = CP_{cl}C' + R \quad (1.200a)$$

$$\Gamma_{cl} = DP_{cl}D' \quad (1.200b)$$

The effect of control can be evaluated by comparing the open loop stationary covariance,  $\Gamma_{ol}$ , of the outputs,  $z_k$ , and the closed loop stationary covariance,  $\Gamma_{cl}$ , of the outputs. The variance may be analyzed in terms of their principal components by a singular-value decomposition or an eigenvalue decomposition of the covariance matrices (chap. 3, Poulsen, 1995a). Using the orthogonality properties of the optimal estimate and its error, it is possible to show  $P_{12} = P_{21} = P_p$  and that  $P_{cl}$  may also be computed directly from the Lyapunov equation

$$P_{cl} = (A + BL)P_{cl}(A + BL)' + \Phi \quad (1.201)$$

in which

$$\begin{aligned} \Phi = & (BL(K_f C - I))P_p(BL(K_f C - I))' \\ & + (A + BL)P_p[BL(K_f C - I)]' + [BL(K_f C - I)]P_p(A + BL)' \\ & + \begin{bmatrix} G & BLK_f \end{bmatrix} \begin{bmatrix} Q & S \\ S' & R \end{bmatrix} \begin{bmatrix} G & BLK_f \end{bmatrix}' \end{aligned} \quad (1.202)$$

□

#### Example 1.4.2 (Closed Loop Properties for a Simple System)

To illustrate the ideas proposed in example 1.4.1, consider the system

$$x_{k+1} = 0.9x_k + u_k + w_k \quad (1.203a)$$

$$y_k = x_k + v_k \quad (1.203b)$$

$$z_k = x_k \quad (1.203c)$$

in which the process and measurement noise,  $w_k$  and  $v_k$ , are distributed as

$$\begin{bmatrix} w_k \\ v_k \end{bmatrix} \in N_{iid} \left( \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} Q & S \\ S' & R \end{bmatrix} \right) \quad \begin{bmatrix} Q & S \\ S' & R \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 0.1 \end{bmatrix} \quad (1.204)$$

The initial state,  $x_0$ , is specified exactly as  $x_0 = 0$ .

An LQG controller for this system with the objective

$$\phi = E \left\{ \frac{1}{2} \sum_{k=0}^{\infty} z'_k Q_z z_k \right\} \quad Q_z = 100 \quad (1.205)$$

is given by

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + 0.91483 (y_k - \hat{x}_{k|k-1}) \quad (1.206a)$$

$$u_k = -0.9 \hat{x}_{k|k} \quad (1.206b)$$

$$\hat{x}_{k+1|k} = 0.9 \hat{x}_{k|k} + u_k \quad (1.206c)$$

For one realization of  $\{w_k, v_k\}$ , the open loop outputs,  $z_k$ , as well as the closed loop inputs,  $u_k$ , and outputs,  $z_k$ , are plotted in figure 1.4. By inspection, it is obvious that the variance of the closed loop systems is much smaller than the variance of the open loop system. Therefore, closed loop operation enables closer operation at a constraint in the outputs,  $z_k$ , with the possibility of lower production costs.

In the limit,  $k \rightarrow \infty$ , the stationary distribution of the outputs,  $z_k$ , in the open loop system is

$$k \rightarrow \infty : \quad z_k \in N(0, \Gamma_{ol}) \quad \Gamma_{ol} = 5.2632 \quad (1.207)$$

In the closed loop situation the corresponding distribution of the outputs,  $z_k$ , is

$$k \rightarrow \infty : \quad z_k \in N(0, \Gamma_{cl}) \quad \Gamma_{cl} = 1.0741 \quad (1.208)$$

The error function,  $\text{erf}(x)$ , is

$$\text{erf}(x) = \begin{cases} \frac{2}{\sqrt{\pi}} \int_0^x \exp(-t^2) dt & x \geq 0 \\ -\frac{2}{\sqrt{\pi}} \int_0^{-x} \exp(-t^2) dt & x < 0 \end{cases} \quad (1.209)$$

and its inverse,  $\text{erfinv}(y)$ , is denoted

$$\text{erfinv}(y) = \text{erf}^{-1}(y) \quad (1.210)$$

A scalar normally distributed variable

$$X \in N(\mu, \sigma^2) \quad (1.211)$$

has the density function

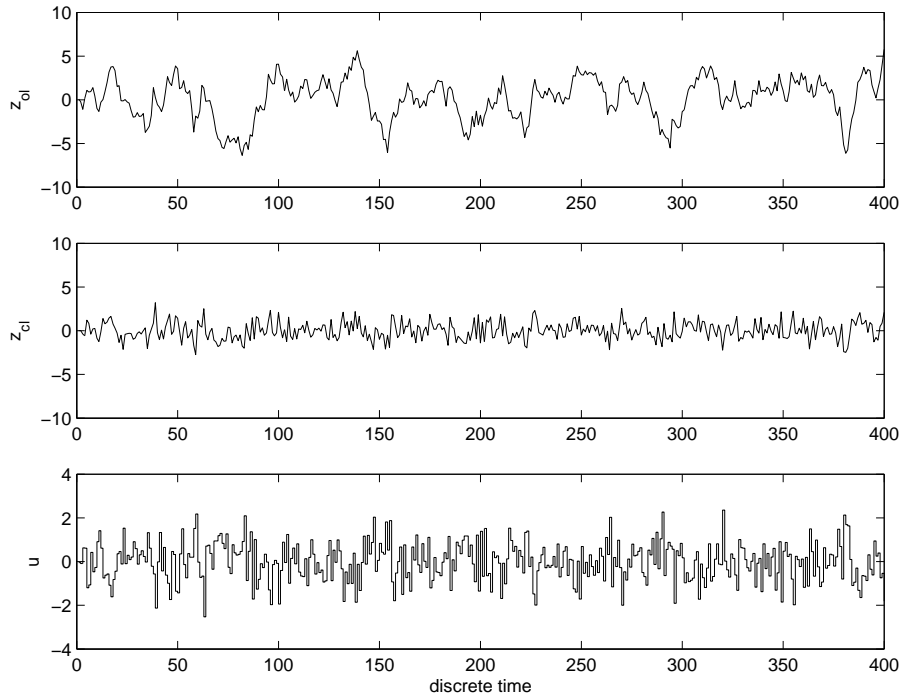
$$\phi(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left( - \left( \frac{x - \mu}{\sqrt{2\sigma^2}} \right)^2 \right) \quad (1.212)$$

Consider the situation, in which the variance,  $\sigma^2$ , of  $X$  is given and the mean,  $\mu$ , is to be determined such that  $X$  is greater than  $x_{\min}$  with probability  $\alpha$ , i.e.

$$P\{X \geq x_{\min}\} = \alpha \quad (1.213)$$

which may be expressed as the condition

$$\int_{-\infty}^{x_{\min}} \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left( - \left( \frac{x - \mu}{\sqrt{2\sigma^2}} \right)^2 \right) dx = 1 - \alpha \quad (1.214)$$



**Figure 1.4.** The effect of control. The upper plot is the open loop output sequence and the middle plot shows the LQG closed loop sequence for the same realization of the process and measurement noise. The lower plot shows the actuator position used in the closed loop situation. The variance of the open loop output sequence is 5.26, while the variance of the closed loop output sequence is reduced to 1.07 by the feedback action.

and is equivalent to

$$\frac{1}{\sqrt{\pi}} \int_{-\infty}^{\frac{x_{\min} - \mu}{\sqrt{2\sigma^2}}} \exp(-t^2) dt = 1 - \alpha \quad (1.215)$$

Therefore,  $\mu$  may be obtained by the expression

$$\mu = x_{\min} - \sqrt{2\sigma^2} \operatorname{erfinv}(1 - 2\alpha) \quad (1.216)$$

Note that the stationary variances of the open and closed loop linear systems are independent of the mean values. They are only dependent on the covariances and the system matrices. Consider the case, in which we relax the output constraint  $z_k \geq -1$  by saying that  $z_k$  should be greater than  $-1$  99% of the time.

In the open loop case, the mean value,  $\bar{z}_{ol}$ , of the output variable should be

$$\bar{z}_{ol} = -1 - \sqrt{2 \cdot 5.2632} \operatorname{erfinv}(1 - 2 \cdot 0.99) = 4.337$$

to satisfy the constraint  $z_k \geq -1$  with a probability of 99%. This is realized by letting  $u_k = 0.4337$ .

In the closed loop case, the mean,  $\bar{z}_{cl}$ , of the output variable,  $z_k$ , should be

$$\bar{z}_{cl} = -1 - \sqrt{2 \cdot 1.0741} \operatorname{erfinv}(1 - 2 \cdot 0.99) = 1.411$$

to satisfy the constraint  $z_k \geq -1$  99% of the time. Consequently, the back-off required in the closed loop case is considerably lower than the back-off required in the open loop case. This would in most cases translate into lower cost of production and be a measure of the economic value generated by the controller.  $\square$

As the preceding examples have demonstrated predictive control has the ability to reduce the output variance and this reduction has a certain economic value. However, whether this variance reduction is possible or not depends on the ratio of process and measurement variance. In addition it is important that covariance information used by the predictor and filter in the controller corresponds to the covariance of the process. Appendix E provides one method to obtain adequate covariance information along with identification of the model. Odelson (2003) provides a recent discussion of covariance estimation.

## 1.5 Thesis Organization

This survey of topics within predictive control provides a solid basis for understanding the main topic of the thesis: Application of the extended linear quadratic optimal control problem and its solution for numerical problems in moving horizon estimation and control. The rest of the thesis is organized as follows.

**Chapter 2** discusses computational methods and inherent approximations in model predictive control. By considering the stochastic optimal control problem, the approximations and assumptions of model predictive control are pinpointed. In an ad hoc fashion the separation principle and certainty-equivalence are assumed to prevail, such that the stochastic optimal control problem may be separated into an estimation problem and a deterministic optimal control problem. Both the estimation problem and the obtained deterministic optimal control problem are demonstrated to be instances of a constrained nonlinear optimal control problem. In the sequential quadratic programming algorithm for solution of constrained nonlinear optimal control problems, the quadratic subproblem generated at each iteration is shown to be a constrained linear-quadratic optimal control problem. Procedures for generation of the constrained linear-quadratic optimal control problem and its data from the nonlinear estimation problem, the nonlinear control problem, the linear moving horizon estimator, and the linear moving horizon controller are provided. The significance of these conversions to constrained linear-quadratic optimal control problems is that the entire model predictive control problem can be solved



efficiently by having efficient algorithms tailored for solution of the constrained linear-quadratic optimal control problem. The major intention in this chapter is to emphasize the central role of the constrained linear-quadratic optimal control problem in model predictive control such that tailored algorithms for the constrained linear-quadratic optimal control problem is motivated and justified.

**Chapter 3.** A primal active set, a dual active set, and an interior point algorithm for solution of the constrained linear quadratic optimal control problem are outlined. The major computational effort in all these algorithms reduces to solution of certain unconstrained linear quadratic optimal control problems, i.e. the extended linear quadratic control problem. A Riccati recursion procedure for effective solution of such unconstrained problems is stated.

**Chapter 4.** Based on dynamic programming, Riccati recursion procedures for the linear-quadratic optimal control problem as well as the extended linear-quadratic optimal control problem are developed. Compared to alternative solution procedures such as control vector parameterization by elimination of the states, the Riccati based procedure is highly efficient for long prediction horizons. The extended linear-quadratic optimal control problem may also be regarded as an equality constrained quadratic program with special structure. The computation of the optimal solution-Lagrange multiplier pair for a convex equality constrained quadratic program is specialized to the extended linear-quadratic optimal control problem treated as a quadratic program. Efficient solution of the highly structured KKT-system corresponding to the extended linear-quadratic optimal control problem is facilitated by the Riccati recursion developed by dynamic programming.

**Chapter 5** presents the principles for efficient solution of unconstrained nonlinear optimal control problems described by ordinary differential equations. These principles are presented through numerical solution of a continuous-time nonlinear optimal control problem of the Bolza form. To focus on the basic principles involved and for illustrative purposes, the continuous-time Bolza problem is discretized by the explicit Euler method. The discrete-time nonlinear optimal control problem of the Bolza form is solved by different SQP methods and an algorithm based on the discrete maximum principle. The SQP algorithms presented are implementations based on open- and closed-loop feasible path control vector parameterizations as well as an infeasible path simultaneous procedure. Two procedures for solution of the quadratic programs are presented. In the first procedure, the structure of the quadratic programs arising in the solution of the nonlinear optimal control problem is utilized by a Riccati iteration based factorization of the resulting KKT-system. In the second procedure, an efficient procedure for elimination of the states and solution of a dense reduced space quadratic program is presented. These methods are compared for a simple process example operated around an unstable equilibrium. The infeasible path and the closed-loop feasible path algorithms converge for this example. The implemented open-loop feasible path algorithms are not able to converge to an unstable equilibrium. The Riccati based solution procedure

enables implementation of the stabilized infeasible path SQP algorithm as well as the closed-loop feasible path SQP algorithm. The methods are presented in a framework that is easily extended to constrained nonlinear optimal control problems. Such extensions and methodologies for efficient integration of the ordinary differential equations as well as the corresponding sensitivity equations are discussed.

**Chapter 6** summarizes the main contribution of this thesis. It briefly discusses the pros and cons of using the extended linear quadratic control framework for solution of deterministic optimal control problems.

**Appendices.** Appendix A demonstrates how quadratic programs arise in sequential quadratic programming algorithms. Appendix B uses a control vector parameterization approach to express various extended constrained linear quadratic optimal control problems as standard quadratic programs. Appendix C discuss construction of maximal output admissible sets. It provides an algorithm for computation of the maximal output admissible set for linear model predictive control. Appendix D provides results concerning linear regression. Appendix E discuss prediction error methods for identification of linear models tailored for model predictive control.



# The Computational Principles of Model Predictive Control

The computational methods and inherent approximations of model predictive control are discussed. By considering the stochastic optimal control problem, the approximations and assumptions of model predictive control are pinpointed. Industrial practiced model predictive is a mathematical tractable but suboptimal solution to the stochastic optimal control problem. In an ad hoc fashion the separation principle and certainty-equivalence are assumed to prevail, such that the stochastic optimal control problem may be separated into an estimation problem and a deterministic optimal control problem. Both the estimation problem and the obtained deterministic optimal control problem are demonstrated to be instances of a constrained nonlinear optimal control problem. In the sequential quadratic programming algorithm for solution of constrained nonlinear optimal control problems, the quadratic subproblem generated at each iteration is shown to be a constrained linear-quadratic optimal control problem. Procedures for generation of the constrained linear-quadratic optimal control problem and its data from the nonlinear estimation problem, the nonlinear control problem, the linear moving horizon estimator, and the linear moving horizon controller are provided. The significance of these conversions to constrained linear-quadratic optimal control problems is that the entire model predictive control problem can be solved efficiently by having efficient algorithms tailored for solution of the constrained linear-quadratic optimal control problem. The major intention in this chapter is to emphasize the central role of the constrained linear-quadratic optimal control problem in model predictive control such that tailored algorithms for the constrained linear-quadratic optimal control problem is motivated and justified.

## 2.1 Introduction

Model predictive control is the predominant paradigm of advanced control in the process industries. An important reason of this dominance is that the model predictive framework is sufficiently versatile such that it readily includes monitoring and fault diagnosis, as well as the classical control topics such as system identification, state estimation, and regulation. Model predictive control provides a systematic and scientific sound methodology, directly aimed at cutting operating costs, improving product quality by variance reduction, and satisfying any environmental and safety constraint of industrial processes. These features along with a very impressive documented industrial track record (c.f. Clarke, 1988; Richalet, 1993; Qin and Badgwell, 1996, 2000) have made model predictive control an established and routinely practiced technology for optimizing industrial processes in several industrial sectors.

Model predictive control consists of a very broad class of control systems that compute the control strategies such that the outcome predicted by a model of the system is as favorable as possible according to some criterion. This concept of model predictive control is sufficiently broad to include almost any classical optimal control scheme. Hence, the scope is limited to focus mainly on model predictive control of constrained systems implemented by online optimization. In fact the main ingredients distinguishing model predictive control from other control techniques is that it explicitly uses a model to predict the future evolution of the system. In current industrial practice, this prediction is used to compute the solution of an open-loop optimal control problem on-line. The solution to this open-loop optimal control problem is a sequence of open-loop optimal control inputs. The first control input in the obtained sequence is implemented on the process and the procedure is repeated as new information becomes available for the controller. This type of control is called moving horizon control as well as receding horizon control.

### 2.1.1 Organization and Purpose

The approximations invoked in implementable moving horizon estimation and control when applied for a stochastic optimal control problem are discussed in this chapter. Contrary to the common naive belief, industrial practiced moving horizon estimation and control is not the best possible controller in a theoretical sense just because the estimator and the controller are posed as optimization problems. The estimator and controller may be optimal for the deterministic problem but that does not make them optimal for the stochastic problem. They are merely suboptimal practical solutions of the stochastic optimal control problem as they account for uncertainty in an ad hoc fashion. The explicit discussion of these approximations suggests how some of the approximations imposed for establishing a real-time computational procedure may be relaxed, modified or lifted to have model predictive controllers with even better perfor-

mance than realized in current industrial practice. The result of this discussion is a statement of the mathematical problems that must be solved in moving horizon estimation and control for suboptimal stochastic control.

While the discussion of the stochastic optimal control problem provides an answer to *which* problems to solve, the issues of *how* to solve then still remain undetermined. The main purpose of this chapter is then to establish and explain the key role of the linear-quadratic optimal control problem in implementation of nonlinear as well as linear moving horizon estimation and control. The linear-quadratic optimal control problem is distinguished from the classic finite horizon LQ problem by the facts that it contains affine terms in the state dynamics and linear terms in the stage costs. These apparently innocent modifications have a tremendous impact as they in a transparent and straightforward way enables numerical solution of essentially all problems within model predictive control by solution of linear-quadratic optimal control subproblems. This central and important role of the linear-quadratic optimal control problem is explained by introducing the problem and then demonstrate how it arises as the quadratic subproblem in the solution of nonlinear optimal control problem. Though efficient solution methods for the linear-quadratic optimal control problem are outlined, the purpose is not to provide a detailed discussion of this topic as this discussion is provided in the remaining part of this dissertation. It is merely to outline how the linear-quadratic optimal control problem arises as a subproblem in the solution of moving horizon estimation and control problems. By this approach one can subsequently focus on the efficient solution of the generic linear-quadratic optimal control problem without regard to its specific origin.

The linear-quadratic optimal control problem is demonstrated as the quadratic programming subproblem for the constrained general nonlinear optimal control problem as well as the constrained least-squares nonlinear optimal control problem when these problems are solved by sequential quadratic programming. Next, the nonlinear moving horizon estimation problem is demonstrated to be an instance of the constrained least-squares nonlinear optimal control problem. Using the same technique as for the constrained nonlinear moving horizon estimator, the constrained linear moving horizon estimator is transformed to a constrained linear-quadratic optimal control problem. The transformation of the moving horizon estimation problems to nonlinear optimal control problems includes a discussion of how to obtain offset free estimates and the influence of such a requirement on the model formulations. This treatment discusses the transformation of the mathematical program constituting the moving horizon estimator to an optimal control problem. However, it does not discuss how to select the horizon nor how to compute the cost-to-arrive matrices. These issues are treated by Rao and Rawlings (2000), Rao (2000), and Tenny (2002).

The nonlinear as well as the linear moving horizon controller are stated as infinite-horizon mathematical programs. By a decomposition, which in general is suboptimal, the solution of the infinite horizon moving horizon controllers

are computed by solution of a static optimization problem (the target problem) and a finite horizon optimal control problem. The solution of the general infinite horizon moving horizon control problem is established heuristically as the solution of a general static optimization problem and a constrained nonlinear optimal control problem. In its full generality, this procedure for obtaining the solution to an infinite horizon optimal control problem is an approximation (c.f. Carlson and Haurie, 1987; Carlson *et al.*, 1991). However, if the horizon of the finite-horizon optimal control problem is selected sufficiently long, the closed-loop solution obtained by this procedure is expected to approximate the actual infinite-horizon solution well due to the moving horizon implementation.

A similar decomposition is described for the case of infinite horizon moving horizon control with linear models. In this case, the finite horizon model predictive control approximation is transformed to a linear-quadratic optimal control problem. However, even in the linear case, the proposed decomposition procedure in which the infinite horizon problem is approximated by a finite horizon problem will only be exact for cases in which the steady-state target computed by the static optimization is in the interior of the maximal output admissible set (c.f. Scokaert and Rawlings, 1998; Gilbert and Tan, 1991). For problems with infeasible targets and the steady-state target being on the boundary of the maximal output admissible set, the linear terms in the stage costs of the constrained linear-quadratic optimal control problem are indispensable in generating good approximations to the infinite-horizon moving horizon control problem.

Finally, the linear terms of the stage costs and the affine terms of the state dynamics distinguishing the linear-quadratic optimal control problem from the classic LQ control problem facilitates anticipatory control. Anticipatory control is a generalization of feed forward control in which the controller at some time is provided information about future values of the set-points as well as future values of the loads. The application of the linear-quadratic optimal control paradigm for such problems is briefly discussed. It also briefly mentioned that the anticipatory controller is instrumental in some realizations of coordinated decentralized model predictive controllers.

### 2.1.2 Literature Review

Advanced control in the process industries as it is practiced today is almost synonymously with model predictive control. The papers by Richalet *et al.* (1976, 1978) and Cutler and Ramaker (1980) have been particularly influential and are often regarded as the origins of model predictive control.

The connections between the closely related optimal minimum time problem and linear programming were recognized first by Zadeh and Whalen (1962). Propoi (1963) proposed the moving horizon approach to this problem. Experimental implementation of the constrained time optimal controllers was first

reported by Knudsen (1975) for a pilot plant evaporator. This development has been reviewed by Gutman (1982) and its connection to model predictive control discovered by Chang and Seborg (1983).

Model predictive control has been reviewed from a theoretical point of view by Garcia *et al.* (1989), Rawlings *et al.* (1994), Lee and Cooley (1997), Morari and Lee (1999), Mayne *et al.* (2000). Froisy (1994) provides a vendor's perspective on industrial MPC technology and summarizes likely future developments. Qin and Badgwell (1996, 2000) describe industrial model predictive control implementations. Rawlings (2000) provides a short tutorial on the essential principles of nonlinear model predictive control. Mayne (1995, 1997) gives an overview of different nonlinear model predictive controllers. The computational principles for solution of nonlinear model predictive control problems have been described by Biegler (1998), Allgöwer *et al.* (1999), and Binder *et al.* (2001a). Clarke (1994), Kantor *et al.* (1997), Allgöwer and Zheng (2000) are conference proceedings devoted to model predictive control and related areas. The collections edited by Bulirsch and Kraft (1994), Henson and Seborg (1997), and Grötschel *et al.* (2001) are related to model predictive control and real-time optimization.

### 2.1.2.1 Numerical Methods

The major operation in model predictive control concerns solution of a discrete-time optimal control problem in real time. The main purpose of this chapter is to formulate the problems in model predictive control as optimal control problems, but not to provide detailed solution algorithms. However, the formulation of the problems to be solved are to some extent determined by the solution algorithms available. We will therefore briefly survey the major algorithms for solution of the deterministic discrete-time optimal control problem. The major algorithm classes for solution of the discrete-time optimal control problem are *maximum principle* algorithms, *differential dynamic programming*, and methods based on *mathematical programming*. In contemporary practice, the optimal control problems are solved by mathematical programming. However, the development of the optimal control theory and mathematical programming took place as separate developments.

One class of algorithms for solution of discrete-time optimal control problems is the maximum principle algorithms. These algorithms are based on the Hamiltonian of the optimal control problem. The classical maximum principle asserts that at the optimal solution, the controls,  $u_k$ , maximize the Hamiltonians and there exist adjoint variable equal to the state derivative of the Hamiltonians at the optimal solution. Pontryagin *et al.* (1962) proposed a solution method for the continuous time optimal control problem using the maximum principle. Shortly after the discovery of the maximum principle algorithm for continuous optimal control problems, a weak maximum principle was developed for discrete-time systems by Katz (1962a,b). The weak maximum



principle algorithm states that the Hamiltonian takes a stationary value with respect to  $u_k$  rather than a maximum. However, the relation to the continuous-time maximum principle algorithm was not straightforward and Fan and Wang (1964) concluded incorrectly that the maximum part of the maximum principle algorithm holds under the same fairly weak conditions as in the continuous case. Horn and Jackson (1965) as well as Jackson and Horn (1965) discussed and concluded correctly that the maximum principle is not generally valid for discrete-time optimal control problems, although it is valid for discrete-time linear systems. Propoi (1965) and Halkin (1966) discussed in geometric terms the existence of adjoint variables. The concept of directional convexity was introduced by Holtzman (1966a,b) and Holtzman and Halkin (1966). They used that to sharpen the formulation concerning existence of adjoint variables at the optimum. The general maximum principle algorithm was presented by Nahorski *et al.* (1984). It applied the optimal value function calculated forward. Vidal (1987) formulated the sufficient maximum principle. Hartl *et al.* (1995) surveys the different maximum principle algorithms for continuous systems. Ravn (1999) provides a nice survey of the development of discrete-time maximum principle algorithms. Everett (1963) provided an interesting interpretation of the maximum principle algorithm for discrete-time systems as a Lagrange decomposition. In retrospect, it is clear that the maximum principle for discrete-time systems is just a special formulation of the KKT first-order conditions. The maximum principle algorithms are special methods for solving these systems of equations.

The second major practical approach to the solution of deterministic discrete-time optimal control problems is differential dynamic programming. Differential dynamic programming is inspired by dynamic programming. Differential dynamic programming proceeds much like the traditional formal dynamic programming algorithm, except that at each stage, the optimal return function from the next stage onward as well as the cost from that stage are replaced by their quadratic approximations about the current controls and states. The curse of dimensionality associated with dynamic programming, is overcome by differential dynamic programming by constructing linear-quadratic approximations of the optimal value function and associated dynamics at each stage of the backward recursion in dynamic programming. The main motivation for applying differential dynamic programming to the nonlinear discrete-time optimal control problem has been that its computational cost is  $O(Nm^3)$  in which  $m$  is the dimension of the controls (i.e.  $u_k \in \mathbb{R}^m$ ) and  $N$  is the horizon. The computational expense in conventional application of Newton's method by elimination of the states is  $O(N^3m^3)$ . However, Jonson (1983), Pantoja (1988), and Dunn and Bertsekas (1989) have independently demonstrated that Newton's method may also be applied to solve the deterministic optimal control problem in computational cost  $O(Nm^3)$ . In this approach, both the states and the controls are treated as decision variables and the transition equations are left as constraints. Differential dynamic programming is not identical with Newton's

method for optimal control problems. The backward recursion using dynamic programming is identical for Newton's method and differential dynamic programming. However, in the forward recursion Newton's method generates the state trajectory using the linearization of the transition equations while differential dynamic programming applies the transition equations themselves for generation of the state trajectory. In that respect, differential dynamic programming may be regarded as a feasible path SQP method for unconstrained optimal control problems. Mayne (1966), Jacobson and Mayne (1970), and Dyer and McReynolds (1970) suggested differential dynamic programming algorithms for locally unconstrained optimal control problems. Ohno (1978a), Murray and Yakowitz (1979), and Yakowitz (1986) applied differential dynamic programming techniques to the solution of constrained optimal control problems. Murray and Yakowitz (1984) examined differential dynamic programming and Newton's method for optimal control problems. They showed that these methods are not identical and that differential dynamic programming has a quadratic rate of convergence. Liao and Shoemaker (1991) provided a more direct proof for the quadratic convergence of differential dynamic programming. Yakowitz and Rutherford (1984) developed computational techniques such that differential dynamic programming to generic optimal control problems. They also provided line search methods and criteria to ensure global convergence. Differential dynamic programming requires first as well as second derivatives of the stage cost as well as the state transition equations. This computational burden was removed by Sen and Yakowitz (1987) and Rakshit and Sen (1990) who devised quasi-Newton methods for differential dynamic programming. The developed stage wise rank-one updates require the first derivatives of the stage cost and the state transition at each stage. Yakowitz (1988, 1989) reviews theoretical and computational aspects of differential dynamic programming. Ravn (1999) provides a recent introduction to differential dynamic programming and compares differential dynamic programming with Newton's method. Practical implementations and applications of differential dynamic programming have been described in Murray and Yakowitz (1979), Yakowitz and Rutherford (1984), Liao and Shoemaker (1991). Ohno (1978b) and Murray and Yakowitz (1981) demonstrated that separable mathematical programs could be formulated as optimal control problems and solved efficiently using differential dynamic programming. In retrospect, differential dynamic programming may be regarded as a feasible path solution method that utilizes a Riccati recursion for discrete-time optimal control problems.

The third technique for solution of optimal control problems is mathematical programming. Currently, mathematical programming is the most popular and commonly applied technique for solution of optimal control problems. However, the development of optimal control theory and mathematical programming took place as separate activities. In the mainstream control community at the early days of the optimal control development, it was not at all obvious how to formulate optimal control problems as mathematical programs. Lee

and Markus (1968), Canon *et al.* (1970), and Tabak and Kuo (1971) are early textbooks treating the optimal control problem as a mathematical program. Luenberger (1972) and Polak (1973) also describes the formulation of optimal control problems as mathematical programs. Recent books dealing with formulation of optimal control problems as mathematical programs includes Bertsekas (1995b), Polak (1997), and Betts (2001). Nocedal and Wright (1999) provide a very nice introduction to the numerical techniques employed in solving mathematical programs. Powell (1977) was the first to present a numerical efficient sequential quadratic programming algorithm for solution of nonlinear programs. The solution of optimal control problems as mathematical programs is typically accomplished using sequential quadratic programming (SQP) algorithms as they require far fewer function evaluations than gradient methods. The main differences of the SQP algorithms applied for solution of optimal control problems concerns 1) how the quadratic program subproblems are solved, 2) whether a line-search or trust region strategy is used to select the step size, and 3) whether the intermediate iterates generated are enforced to be feasible or allowed to be infeasible such that only the final optimal iterate is feasible. The quadratic subproblems may be solved by active set algorithm or interior-point algorithms. Both active set and interior-point algorithms may utilize the structure of the linear-quadratic optimal control problem or they may be dense quadratic programs obtained after elimination of the states. The solution of the quadratic program corresponding to the linear-quadratic optimal control problem has been discussed by Wright (1996) and Allgöwer *et al.* (1999).

Jonson (1983), Pantoja and Mayne (1989), and Dunn and Bertsekas (1989) independently solved unconstrained discrete-time optimal control problems utilizing the special structure of the control problem. The resulting solution procedure is based on Riccati recursions such that the linear-quadratic optimal control problem structure is utilized. Dohrmann and Robinett (1997) extended the method of Dunn and Bertsekas (1989) to discrete-time optimal control problems with equality constraints and allowed for free final time as well.

Wright (1993), Steinbach (1994, 1995), Rao *et al.* (1998), and Tenny (2002) solved constrained discrete-time optimal control problems by interior-point methods. By retaining the states as decisions variables and leaving the state transition equations as constraints, the essential linear algebra operation in the interior-point algorithms is the just mentioned Riccati recursion for solving unconstrained linear-quadratic optimal control problems. By this method the computational cost of solving the optimal control problem by mathematical programming is  $O(Nm^3)$ . The alternative methods based on state elimination has computational complexity  $O(N^3m^3)$ .

Wright and Tenny (2002) developed a feasible path trust-region sequential quadratic programming algorithm for general nonlinear programs. Tenny *et al.* (2002) applied that algorithm for the solution of nonlinear optimal control problems in a model predictive control context using a Riccati iteration based

interior point algorithm. Tenny (2002) describes application of this algorithm to moving horizon estimation and control problems. In a model predictive control context, the advantage of feasible path methods is that the algorithm may terminate early due to the real-time constraint and still provide a feasible solution to be implemented. As discussed by Scokaert *et al.* (1999) such methods are stabilizing as reduction of the value function is sufficient for stability. Leineweber (1995, 1999), Bock *et al.* (2000), and Diehl *et al.* (2001) have developed a multiple shooting algorithm for nonlinear optimal control problems. This algorithm employs a line-search methodology to ensure global convergence. The multiple-shooting algorithm is an infeasible path approach as the state transition equations are enforced within each shooting interval but not between the shooting intervals. Diehl *et al.* (2002) applied this algorithm in nonlinear model predictive control of a distillation column. Biegler (1998, 2000) provides an overview of optimization techniques for dynamic optimization. General sparse equation solvers are considered for the solution of the structured optimality conditions for the dynamic optimization problem.

Vassiliadis *et al.* (1994a,b) and Feehery and Barton (1998) studied continuous time optimal control problems with the purpose of generating optimal trajectories.

Binder *et al.* (2000, 2001b,c) consider iterative adaptive refinement of the mesh on which the nonlinear optimal control and estimation problem are solved. By this procedure the mesh is fine in time intervals at which the trajectories change drastically and coarse in other intervals.

### 2.1.2.2 Control Approaches

Industrial implementations of model predictive control has typically invoked the separation principle as well as the certainty equivalence principle. In this framework, the regulation problem may be treated as a deterministic problem. Feedback has either been obtained by assuming a constant output disturbance as the cause of any mismatch between predictions and observed outputs or by using a Kalman type filter (c.f. Qin and Badgwell, 1996, 2000; Allgöwer *et al.*, 1999). By this approach, it is tacitly assumed that the system is essentially deterministic and that the effect of any uncertainties can be efficiently eliminated by the use of feedback. The types of optimal control has been described by Dreyfus (1964). Arrow *et al.* (1949) and Arrow (1951) discuss various probabilistic and min-max criteria in optimal decision making. Kalman (1960a,b) and Kalman and Bertram (1960) developed the principles for optimal control of linear-quadratic systems. The work by Kalman is a basis for unconstrained optimal control of linear-quadratic systems.

Loeblein and Perkins (1999a,b) address the effect of uncertainties on the steady-state economics on model predictive controller using the concept of back-off. Schwarm and Nikolaou (1999) treat the output constraints as chance con-

straints. By this method they implicitly back-off from the optimal deterministic steady-state operating point. Kassmann *et al.* (2000) incorporate the uncertainty description in the formulation of the target problem and show that the resulting problem is a second order cone program. The second order cone program is efficiently solved using primal-dual interior point algorithms.

Bar-Shalom (1981) considers the stochastic optimal control problem and apply the concepts of *caution* and *probing* introduced by Feldbaum (1965) to demonstrate that the objective of a stochastic optimal control problem is due to a deterministic effect, an effect due to uncertain parameters, and an effect due to external stochastic disturbances. Lee and Cooley (1998) derive the optimal controllers linear systems in which the parameters are uncertain and independently distributed. Bitmead *et al.* (1990) provide a description of adaptive model based predictive controllers in the framework of input-output models (see also Clarke, 1994). Dollar *et al.* (1993) provide a description of an industrial adaptive model predictive controller.

Eaton and Rawlings (1992) and Mosca and Casavola (1995) discuss an extension of model predictive control in which future set-point trajectories and disturbances are provided to the controller. This form of control is a generalization of feed forward control and is called *anticipatory* control.

## 2.2 Model Predictive Control

Various model predictive controllers for solution of the stochastic optimal control problem will be discussed. These controllers differ by the way they treat the uncertainties of the prediction due to the stochastics of the system and whether they treat the parameters of the system as perfectly known or as uncertain and specified by some stochastic distribution.

The dynamic programming optimal controller reside in one end of this spectrum of controllers discussed. It handles the uncertainties of the predictions in a stochastic optimal way. The mathematical ideality of this controller comes at the price of very expensive and demanding computations for almost all practical problems. Therefore, the dynamic optimal controller is hardly implementable for any practical system. The unconstrained linear-quadratic Gaussian stochastic optimal control problem is the much celebrated exception.

At the other extreme of this spectrum is the moving horizon estimators and controllers that are typically considered for industrial model predictive control. Motivated by the stochastic optimal *separation* of estimation and control for unconstrained linear-quadratic Gaussian systems and the *certainty equivalence* principle, this type of control of nonlinear stochastic systems decompose the computations into estimation of the current state and computation of the future control input values by solution of deterministic optimal control problem in which the stochastic variables are replaced by their mean values.

### 2.2.1 Stochastic Optimal Control

To understand the approximations inherent in the model predictive controller typically applied in the process industries, we consider a discrete-time stochastic optimal control problem with imperfect information. For simplicity let the horizon of this stochastic control problem be finite, say  $N$ .

The dynamics of the system considered is determined by the stochastic difference equation

$$\mathbf{x}_{k+1} = g_k(\mathbf{x}_k, u_k, \mathbf{w}_k) \quad k = 0, 1, \dots, N-1 \quad (2.1)$$

in which  $\{\mathbf{w}_k\}_{k=0}^{N-1}$  is a sequence of independent stochastic variables.  $\mathbf{x}_k$  is the state of the system and  $u_k$  is the controlled input to the system. These inputs are restricted to be in a non-empty possibly state-dependent subset

$$u_k \in U_k(x_k) \quad (2.2)$$

This type of constraint is sufficiently general to include constraints of the type  $h_k(x_k, u_k)$  for  $k = 0, 1, \dots, N-1$  and  $h_N(x_N)$ . Let the measurement equation be the stochastic equation

$$\mathbf{y}_k = \varphi_k(\mathbf{x}_k) + \mathbf{v}_k \quad k = 0, 1, \dots, N-1 \quad (2.3)$$

in which  $\{\mathbf{v}_k\}_{k=0}^{N-1}$  is a sequence of stochastic independent variables representing the measurement noise. Let the set of collected measurements,  $\mathcal{Y}_k$ , and the set of applied inputs,  $\mathcal{U}_k$ , be defined as

$$\mathcal{U}_k = \{u_0, u_1, \dots, u_k\} \quad \mathcal{U}_{-1} = \emptyset \quad (2.4a)$$

$$\mathcal{Y}_k = \{y_0, y_1, \dots, y_k\} \quad (2.4b)$$

Then the information available to the controller,  $\mu_k$ , at stage  $k$  is

$$\mathcal{I}_k = \{\mathcal{U}_{k-1}, \mathcal{Y}_k\} \quad (2.5)$$

The controller,  $\mu_k$ , at stage  $k$  is a function that maps the information  $\mathcal{I}_k$  into an admissible process input

$$u_k = \mu_k(\mathcal{I}_k) \quad (2.6)$$

Let  $\pi$  denote the sequence of controllers, i.e. functions, mapping the information available into a process input

$$\pi = \{\mu_0(\mathcal{I}_0), \mu_1(\mathcal{I}_1), \dots, \mu_{N-1}(\mathcal{I}_{N-1})\} \quad (2.7)$$

The controllers  $\pi$  are selected among the admissible controllers such that the expected cost

$$\phi = E_{x_0, w_k, v_k, k=0,1,\dots,N-1} \left\{ \sum_{k=0}^{N-1} f_k(\mathbf{x}_k, \mu_k(\mathcal{I}_k), \mathbf{w}_k) + f_N(\mathbf{x}_N) \right\} \quad (2.8)$$

is minimized. The optimal sequence of controllers  $\pi$  of this problem may be determined by dynamic programming (c.f. Bertsekas, 1995a). It is important to observe that this optimization is conducted over function spaces and that the sequential availability of information is accounted for by the definition of the controllers  $\mu_k(\mathcal{I}_k)$ . The resulting functions  $\mu_k(\mathcal{I}_k)$  are called the *optimal feedback* controllers.

**Algorithm 1 (Optimal Feedback Controller)**

The control input applied by the optimal feedback controller at each time  $k$  is determined by

1. Let the information vector  $\mathcal{I}_k = \{\mathcal{Y}_k, \mathcal{U}_k\}$  be given and note that this vector may be considered as a state of the stochastic system that evolves according to

$$\mathcal{I}_{k+1} = (\mathcal{I}_k, u_k, y_{k+1}) \quad (2.9)$$

2. Compute

$$V_{N-1}(\mathcal{I}_{N-1}) = \min_{u_{N-1} \in U_{N-1}} [E_{x_{N-1}, w_{N-1}} \{f_{N-1}(x_{N-1}, u_{N-1}, w_{N-1}) + f_N(g_{N-1}(x_{N-1}, u_{N-1}, w_{N-1})) | \mathcal{I}_{N-1}, u_{N-1}\}] \quad (2.10)$$

and set  $\mu_{N-1}(\mathcal{I}_{N-1}) = u_{N-1}^*$  for all  $\mathcal{I}_{N-1}$ . For  $i = N-2, N-3, \dots, k+1, k$  compute recursively

$$V_i(\mathcal{I}_i) = \min_{u_i \in U_i} [E_{x_i, w_i, y_{i+1}} \{f_i(x_i, u_i, w_i) + V_{i+1}(\mathcal{I}_i, u_i, y_{i+1}) | \mathcal{I}_i, u_i\}] \quad (2.11)$$

for all possible future information vectors,  $\mathcal{I}_i$ .

3. Apply the control:  $u_k = \mu_k(\mathcal{I}_k)$

For general problems in which the states are discretized, the complexity of computing the optimal feedback controllers grows exponentially with the number of states, i.e. it has at least complexity  $O(Nd^n)$  in which  $d$  is the number of discretization intervals and  $n$  is the state dimension. Hence, in practice this approach is in general not feasible. Bellman and Dreyfus (1962) called this the curse of dimensionality. The case in which the objective function is quadratic, the state transition equation is linear, and no further constraints exist is a much celebrated exception. For this system an analytical solution exists. The optimal controller is a composite function,  $\mu_k(\mathcal{I}_k) = \mu_k(E\{\mathbf{x}_k | \mathcal{I}_k\}) = \mu_k(\hat{x}_{k|k})$  in which  $\hat{x}_{k|k} = E\{\mathbf{x}_k | \mathcal{I}_k\}$ . The inner function,  $\hat{x}_{k|k} = E\{\mathbf{x}_k | \mathcal{I}_k\}$ , is the conditional mean state which is an optimal estimation of the state. The outer function,  $\mu_k(\hat{x}_{k|k})$ , is an optimal solution to the equivalent optimal stochastic control problem in which perfect state information prevails. This separation of the stochastic optimal control problem with imperfect state estimation to an optimal estimation problem and a stochastic optimal control problem with perfect state information is known as the *separation principle* (c.f. Striebel, 1965; Wonham, 1968; Witsenhausen, 1971). Furthermore, as observed by Simon (1956) (see also Theil, 1957; Joseph and Tou, 1961; Gunckel and Franklin,

1963), the solution of the stochastic linear-quadratic optimal control problem with perfect state information is equivalent to the deterministic linear quadratic optimal control problem in which the stochastic process disturbances  $\mathbf{w}_k$  are replaced by their mean values  $\bar{w}_k = E\{\mathbf{w}_k\}$ . This is called the *certainty equivalence principle*. Consequently, the stochastic linear-quadratic optimal control problem may be solved exactly by solving an optimal linear quadratic estimation problem and a deterministic linear-quadratic optimal control problem.

### 2.2.2 Suboptimal Controllers

The properties of the optimal controller for linear-quadratic stochastic systems have guided the construction of suboptimal controllers for the general stochastic optimal control problem. The separation principle is applied in the computation of most suboptimal controllers such that the controller is decomposed into a state estimation problem and an optimal control problem.

The *certainty equivalent controller* estimates the state and applies at each stage the control that would be optimal if the stochastic quantities were fixed at their expected values. It acts as if a form of the certainty equivalence principle applies. Dreyfus (1964) call this controller the *open-loop-optimal feedback* controller while Bertsekas (1976) refer to it as the *naive feedback* controller as well as the certainty equivalent controller.

#### Algorithm 2 (CEF Controller)

The control input  $u_k = \tilde{\mu}_k(\mathcal{I}_k)$  applied by the certainty equivalent controller at each time  $k$  is determined by

1. Given the information vector  $\mathcal{I}_k$ , compute  $\hat{x}_{k|k} = E\{\mathbf{x}_k|\mathcal{I}_k\}$ .
2. Solve the deterministic optimal control problem

$$\min \tilde{\phi} = \sum_{i=k}^{N-1} f_i(x_i, u_i, \bar{w}_i) + f_N(x_N) \quad (2.12a)$$

$$s.t. \quad x_k = \hat{x}_{k|k} \quad (2.12b)$$

$$x_{i+1} = g_i(x_i, u_i, \bar{w}_i) \quad i = k, \dots, N-1 \quad (2.12c)$$

$$u_i \in U_i(x_i) \quad (2.12d)$$

for  $\{\tilde{u}_k, \tilde{u}_{k+1}, \dots, \tilde{u}_{N-1}\}$  using  $\bar{w}_i = E\{\mathbf{w}_i\}$ .

3. Apply the control input:  $\tilde{\mu}_k(\mathcal{I}_k) = \tilde{u}_k$ .

Step (1) of the certainty equivalent controller computes online an optimal estimate of the state. Step (2) solves online a deterministic optimal control problem and applies the first control action  $\tilde{u}_k$ . The stated algorithm for the certainty equivalent controller computes online the control action  $\tilde{u}_k$  for the state  $\hat{x}_{k|k}$  actually estimated. This is in contrast to offline computation of a control law, i.e. a function, for all possible states. However, this is an implementation



related matter and it is for some systems both possible and preferable to compute the optimal controller,  $\tilde{\mu}_k(\mathcal{I}_k) = \mu_k^d(\hat{x}_{k|k})$ , offline. The controller  $\mu_k^d$  may be obtained by solving the deterministic optimal control problem (2.20) using dynamic programming or sensitivity analysis for constrained linear-quadratic systems (c.f. Bertsekas, 1995a; Bemporad *et al.*, 2002). The conception and justification of the certainty equivalent controller dates to the origins of feedback theory when feedback was employed as a device that compensated for uncertainties and noise in the system. The classic design of feedback controllers tacitly assumes away uncertainties of the stochastic system by fixing stochastic variables at their typical values, and compute the control law based on the deterministic system and considerations such as stability, optimality, and robustness. Like the classic feedback controller, the certainty equivalent controller does not explicitly take uncertainties into account but relies on the feedback mechanism to compensate for uncertainties and noise in the system. For unconstrained linear-quadratic systems, the certainty equivalent controller is optimal. However, for general systems this is a suboptimal feedback controller, but in practice it typically performs well and gives near-optimal policies. Thau and Witsenhausen (1966) give an interesting example in which the certainty equivalent controller has worse performance than the open-loop controller receiving no feedback but taking the uncertainties into account. Witsenhausen (1969, 1970) provides results about how well the cost of suboptimal controllers approximate the optimal feedback controller. Malinvaud (1969) provides a sensitivity property for the certainty equivalent controller.

The *open-loop-optimal* (OLO) controller and the *open-loop-optimal feedback* (OLOF) controller are similar to the certainty equivalent controller and the certainty equivalent feedback controller, respectively. The open-loop-optimal controller receives no feedback, while the open-loop-optimal feedback controller receives feedback at each stage and computes open-loop-optimal control laws based on the information received. The difference between the certainty equivalent controllers and the open-loop-optimal controllers is that the open-loop-optimal controllers take into account the uncertainty about the initial state and process noise in the computation the control laws. The open-loop-optimal feedback controller considers the uncertainties of  $\hat{\mathbf{x}}_{k|k}$ ,  $\mathbf{w}_k, \mathbf{w}_{k+1}, \dots, \mathbf{w}_{N-1}$  when computing the control law  $\bar{\mu}_k(\mathcal{I}_k)$  to be applied at time  $k$ .

### Algorithm 3 (OLOF Controller)

The control input  $u_k = \bar{\mu}_k(\mathcal{I}_k)$  applied by the open-loop-optimal feedback controller at each time  $k$  is determined by

1. Given the information vector  $\mathcal{I}_k$ , compute the conditional probability distribution  $P(x_k|\mathcal{I}_k)$ .
2. Let

$$\bar{\phi} = E_{x_k, \{w_i\}_{i=k, \dots, N-1}} \left\{ \sum_{i=k}^{N-1} f_k(\mathbf{x}_i, u_i, \mathbf{w}_i) + f_N(\mathbf{x}_N) | \mathcal{I}_k \right\} \quad (2.13)$$

Compute a control sequence  $\{\bar{u}_k, \bar{u}_{k+1}, \dots, \bar{u}_{N-1}\}$  that solves

$$\min \bar{\phi} \quad (2.14a)$$

$$s.t. \quad \mathbf{x}_{i+1} = g_i(\mathbf{x}_i, u_i, \mathbf{w}_i), i = k, \dots, N-1 \quad (2.14b)$$

$$u_i \in U_i(\mathbf{x}_i), \quad i = k, \dots, N-1 \quad (2.14c)$$

3. Apply the control input

$$\bar{\mu}_k(\mathcal{I}_k) = \bar{u}_k \quad (2.15)$$

The open-loop-optimal feedback uses the measurements  $\mathcal{Y}_k$  to compute the conditional probability distribution  $P(\hat{\mathbf{x}}_k|\mathcal{I}_k)$ . However, the open-loop-optimal feedback controller selects the control input  $\bar{\mu}_k(\mathcal{I}_k) = \bar{u}_k$  as if no further measurements will be received. In contrast, the optimal feedback controller,  $\mu_k(\mathcal{I}_k)$ , obtained by dynamic programming takes into account that the controller will receive further measurements in the future. A comparison of the certainty equivalent feedback controller, the open-loop-optimal feedback controller, and the optimal feedback controller reveals: 1) the certainty equivalent feedback controller is too aggressive because it neglects uncertainties and assumes perfect information about the future, 2) the open-loop-optimal feedback controller is too cautious as it accounts for the uncertainties but neglects the information received in the future, and 3) the optimal feedback controller strikes the right balance of caution as it considers the uncertainties and takes into account that new information will be received in the future and thus reducing the uncertainty. The control problem (2.28) of the open-loop-optimal feedback controller may be solved using stochastic programming (c.f. Birge and Louveaux, 1997). The practical problem with such an approach is that the stochastic variables must be discretized and the size of the resulting mathematical program becomes easily very large as it grows exponentially with the number of independent stochastic variables.

### 2.2.3 State Estimation

The certainty equivalent feedback controller assumes that the expected state conditioned on the current information can be computed,  $\hat{x}_{k|k} = E\{\mathbf{x}_k|\mathcal{I}_k\}$ , and the open-loop-optimal feedback controller requires the entire conditional probability distribution,  $P(x_k|\mathcal{I}_k)$ . Except for special cases, computation of the conditional expectation and the conditional probability distribution requires the conditional probability density distribution,  $p(x_k|\mathcal{I}_k)$ . As explained by Jazwinski (1970), computation of the conditional probability density involves solving the Chapman-Kolmogorov equation in the discrete-time case and solution of Kolmogorov's forward equation (also known as the Fokker-Planck equation) when the underlying system is continuous and described by stochastic differential equations. In general, solution of either the Chapman-Kolmogorov equation or Kolmogorov's forward equation is a formidable task which can seldom be conducted and certainly not in real-time.

In principle, Monte Carlo simulation is one alternative to the probabilistic approach of state estimation, i.e. computation of the exact evolution of the conditional probability density. In the Monte Carlo approach, a large collection of realizations of the independent stochastic variables are generated by a random number generator, and the corresponding realizations of the dependent state variables are computed and used to construct an approximate conditional probability density. This approximate conditional probability density can be used for computing the conditional mean state as well as the conditional covariance. Even though the computational demand is reduced compared to the exact probabilistic approach, the requirement of a large number of simulation samples limits the practical use of the Monte Carlo method for real-time state estimation.

The third and most practical method for state estimation is the statistical method, which unfortunately is a suboptimal procedure. In this method, the state estimation problem is formulated as a deterministic optimization problem that given the information  $\mathcal{I}_k = \{\mathcal{Y}_k, \mathcal{U}_{k-1}\}$  selects the states  $\{x_i\}_{i=0}^k$ , the process noise  $\{w_i\}_{i=0}^{k-1}$ , and the measurement noise  $\{v_i\}_{i=0}^k$  such that state transition equations and the measurement equations are satisfied, and such that some objective function,  $\psi$ , is minimized

$$\min_{x_i, w_i, v_i} \psi \quad (2.16a)$$

$$s.t. \quad x_{i+1} = g_i(x_i, u_i, w_i) \quad i = 0, \dots, k-1 \quad (2.16b)$$

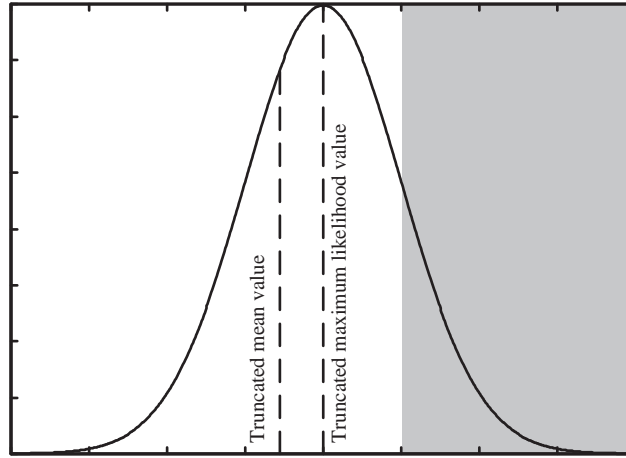
$$y_i = \varphi_i(x_i) + v_i \quad i = 0, \dots, k \quad (2.16c)$$

In a pure statistical approach, no probabilistic information concerning the distribution of  $\mathbf{x}_0$ ,  $\{\mathbf{w}_i\}_{i=0}^{k-1}$ , and  $\{\mathbf{v}_i\}_{i=0}^k$  is available and the selection of the objective function is somewhat arbitrary. Originated by Legendre and Gauss, the objective function has in such situations been selected as a quadratic function such that (2.16) becomes a least-squares problem minimizing the error between the observations and the predicted observations. Later it has been extended such that it is the deviation of the independent variables  $x_0$ ,  $\{w_i\}_{i=0}^{k-1}$ , and  $\{v_i\}_{i=0}^k$  from some nominal variables  $\bar{x}_0$ ,  $\{\bar{w}_i\}_{i=0}^{k-1}$ , and  $\{\bar{v}_i\}_{i=0}^k$  that is minimized. One such objective function is

$$\begin{aligned} \psi = & \frac{1}{2}(x_0 - \bar{x}_0)'P_0^{-1}(x_0 - \bar{x}_0) \\ & + \frac{1}{2} \sum_{i=0}^{k-1} (w_i - \bar{w}_i)'Q_i^{-1}(w_i - \bar{w}_i) \\ & + \frac{1}{2} \sum_{i=0}^k (v_i - \bar{v}_i)'R_i^{-1}(v_i - \bar{v}_i) \end{aligned} \quad (2.17)$$

For this particular objective the weights,  $P_0$ ,  $\{Q_i\}_{i=0}^{k-1}$ , and  $\{R_i\}_{i=0}^k$ , are arbitrary and the function constructed such that the overall weight matrix is

block-diagonal. The state part of the solution of (2.16) is denoted  $\{\hat{x}_{i|k}\}_{i=0}^k$  and called the smoothed state estimate (c.f. Meditch, 1973; Kailath, 1975).  $\hat{x}_{k|k}$  is called the filtered state and is an approximation of the conditional mean state needed in step (1) of the certainty equivalent feedback controller. The optimization problem (2.16) with the objective (2.17) may under the assumption of  $\mathbf{x}_0 \in N(\bar{x}_0, P_0)$ ,  $\mathbf{w}_i \in N(\bar{w}_i, Q_i)$ ,  $\mathbf{v}_i \in N(\bar{v}_i, R_i)$ , and stochastic independence of these stochastic variables be interpreted as selection of the estimate  $\{\hat{x}_{i|k}\}_{i=0}^k$  such that the conditional density,  $p(\{x_i\}_{i=0}^k | \mathcal{I}_k)$ , is maximized. Hence, with proper selection of the objective function the statistical method with a least squares objective corresponds to computation of a maximum likelihood estimate. In a Bayesian sense this is the *maximum a posteriori* estimate as *a priori* information about the means, i.e.  $\bar{x}_0$ ,  $\{\bar{w}_i\}_{i=0}^{k-1}$ , and  $\{\bar{v}_i\}_{i=0}^k$ , is utilized and the covariances are assumed known exactly. When the probability density function is symmetric, the maximum likelihood estimate is identical with the corresponding mean value and minimum variance estimate of the stochastic variable. In contrast, when the probability density function is a non-symmetric truncation of the Gaussian probability density, the maximum likelihood estimate does not have any other probabilistic properties than being the most likely realization of the stochastic variable being estimated. This difference between the mean value and the maximum likelihood estimate is illustrated in figure 2.1 for a stochastic variable with a truncated Gaussian probability density function. Further, in the case of non-Gaussian distribution of any of the exogenous stochastic variables, i.e.  $\mathbf{x}_0$ ,  $\{\mathbf{w}_i\}_{i=0}^{k-1}$ , or  $\{\mathbf{v}_i\}_{i=0}^k$ , the objective function (2.17) gives no other significance to (2.16) than being a least-squares objective (c.f. Deutsch, 1965; Jazwinski, 1970). To retain the maximum likelihood interpretation of the estimated states in the case of non-Gaussian exogenous stochastic variables, the objective function of (2.16) may be selected as the joint probability density function corresponding to the distribution of the exogenous stochastic variables. Like the linear-quadratic control problem, the Gaussian linear-quadratic estimation problem has some special features that may be guiding principles for construction of suboptimal estimators for general nonlinear systems. The particular feature of linear systems with Gaussian external uncertainties, i.e.  $\mathbf{x}_0$ ,  $\{\mathbf{w}_i\}$ , and  $\{\mathbf{v}_i\}$  are stochastic normally distributed variables, is that all stochastic variables in the linear model are normally distributed. The probability density function and the conditional probability density function of joint normally distributed stochastic variables are completely characterized by their mean value and their covariance. Furthermore, by 1) the Gaussian interpretation of  $\mathbf{x}_0$ ,  $\{\mathbf{w}_i\}$ , and  $\{\mathbf{v}_i\}$ , 2) linearity of the state transition equation and the linearity of the measurement equation, and 3) under the assumption of stochastic independence of  $\mathbf{x}_0$ ,  $\{\mathbf{w}_i\}$ , and  $\{\mathbf{v}_i\}$ , the conditional mean value of the states and the corresponding conditional covariances may be computed by simple recursions (c.f. Kailath *et al.*, 2000). The recursive formulas for covariances may be extended to the nonlinear case to provide some ad hoc approximation for the dispersion of the estimated states.



**Figure 2.1.** Truncated normal probability density for illustrating the difference between the truncated mean value and the truncated maximum likelihood estimate.

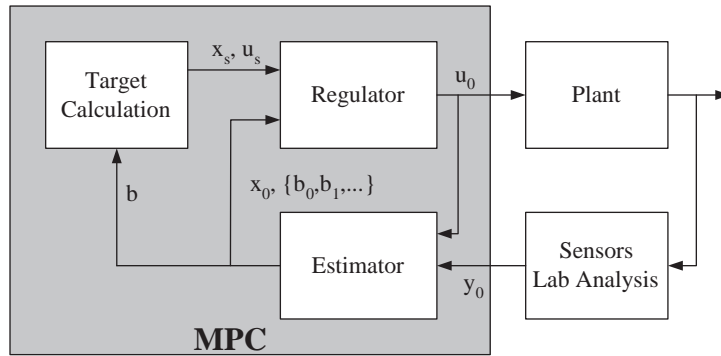
Sayed (2001) discuss state estimation in linear models with uncertain parameters and provide recursions in 1) time- and measurement-update form, 2) prediction form, and 3) information form (see also Hassibi *et al.*, 1999). The structure of these recursions is similar to the corresponding forms of the Kalman filter.

#### 2.2.4 Moving Horizon Estimation and Control

Except for the optimal feedback controller, the suboptimal controllers for stochastic systems employ the separation principle, such that the control system is decomposed into an estimator and a controller. This is illustrated in figure 2.2 for a model predictive controller. The controller in the model predictive controller studied in this paper and illustrated in figure 2.2 consists of a target calculator and a regulator. The estimator may in principle be any of the estimators described in the preceding section. However, to make the implementation practically feasible for non-trivial processes the estimator considered is a least squares estimator (2.16) and (2.17) with additional inequality constraints on the states,  $\{x_i\}_{i=0}^k$ , and the process noise,  $\{w_i\}_{i=0}^{k-1}$ . The controller employed assumes certainty equivalence and may consequently be characterized as a certainty-equivalent feedback controller. When new measurements become available to the estimator, it computes a smoothed estimate of the states  $\{\hat{x}_{i|k}\}_{i=0}^k$  and the exogeneous inputs  $\{\hat{w}_{i|k}\}_{i=0}^k$ . The filtered state,  $\hat{x}_{k|k}$ , is applied as an approximation of the conditional mean and used by the regulator. In addition to estimation of states of the system being controlled, the state es-

estimator may also estimate the states of a disturbance model in order to get an estimate of the influence of past, current, and future exogenous variables on the system (c.f. Muske and Badgwell, 2002). It is assumed that the exogenous variables approaches a limit. This limit is used by the target calculator while the regulator applies the entire trajectory of exogenous variables. The target calculator was introduced by Muske and Rawlings (1993a) to handle control problems with infinite horizons. It computes the optimal steady state, assuming such a state exists, by solving a constrained static optimization problem. The regulator solves a constrained dynamic optimization problem (2.20) and obtains an optimal certainty-equivalent trajectory. The computed open-loop control trajectory is applied on the process until new information become available. By the arrival of new information the entire sequence of calculations is repeated.

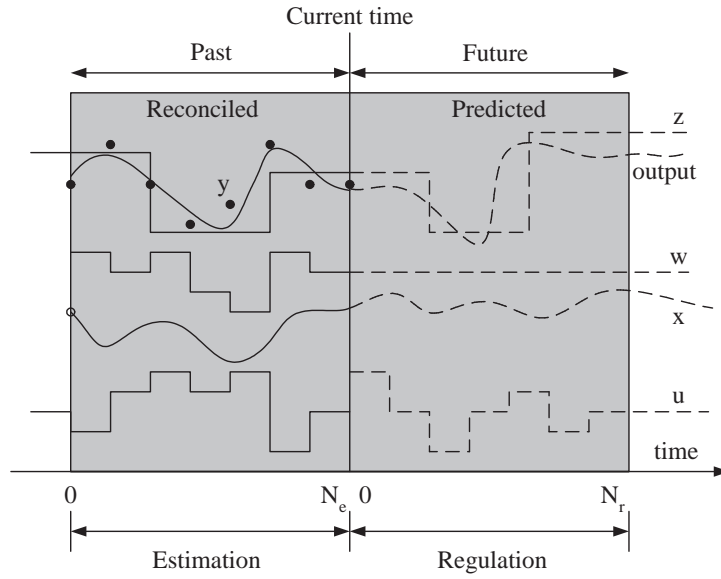
Motivated by mathematical tractability for non-trivial practical processes, the model predictive controller applies the separation principle. The estimator is a least-squares estimator while the controller is a certainty equivalent controller. For continuous processes the horizon,  $N$ , is extremely long and often conceptualized as being infinite. The size of the estimation problem (2.16) grows linearly with the time index and as this index becomes large the estimation problem becomes large. With an infinite horizon, or at least an extremely large finite horizon, the control problem (2.20) is also very large. Both the estimation problem and the control problem are constrained optimization problems which must be solved in real-time. To accomplish that, it is desirable that their size is as small as possible. Hence, yet another approximation is introduced to limit the size of the mathematical programs constituting the estimator and regulator, respectively. The estimator considers only a limited horizon of past data, while the regulator considers only a limited horizon into the future. This is illustrated in figure 2.3 and bounds the size of the mathematical programs that



**Figure 2.2.** Block diagram of a model predictive controller.

must be solved in real-time. As new information becomes available the horizon of the estimator and the regulator is moved as illustrated by 2.4. Therefore, this estimation and control methodology is called moving horizon estimation and control.

In figure 2.3 and figure 2.4, the estimation and the regulation horizons are fixed. In practice, they are often selected adaptively such that the solutions of the mathematical programs of limited size approximates the solution of the corresponding full horizon problems. Further, the information in the past data outside the estimation window is not discarded but approximated by a cost-to-arrive function, just as a cost-to-go function is used to approximately account for the future outside the regulator window (c.f. Muske and Rawlings, 1993a; Scokaert and Rawlings, 1998; Rao and Rawlings, 1999, 2000; Tenny and Rawlings, 2002; Pannocchia *et al.*, 2002). To have good performance and stability of the model predictive controller implemented in a moving horizon fashion it is often necessary to apply quite long estimation- and regulation-horizons. This implies that the size of the mathematical programs constituting the estimator and the regulator may be large even though their sizes are bounded and reduced by the moving horizon approximation.



**Figure 2.3.** Moving horizon estimation and control.



Tacitly, we have assumed that the models considered and their parameters are known precisely. This is never the case for practical control systems in which the parameters,  $\theta$ , of the model are determined by estimation of noisy measurements and therefore uncertain. Such a model may be described as

The parameter vector  $\theta$  is assumed to be fixed and deterministic in the true model of the system. However, the parameter vector in the model used by the controller is a stochastic parameter as it is estimated. The model with an unknown parameter may be formulated as model the standard state space format by augmenting the state vector with the parameters. The resulting model equivalent to (2.18) becomes

$$\theta_{k+1} = \theta_k \quad k = 0, 1, \dots, N-1 \quad (2.19b)$$

with  $\theta_0 = \theta$ . As will be demonstrated, the advantage of the formulation (2.19) compared to (2.18) is that it has the state space structure. Therefore, the case of uncertain models may just be considered as a particular example of a stochastic optimal control problem (c.f. Bertsekas, 1976, 1995a). However, features such as *probing* and *caution* are illustrated and explained particularly well using the uncertain model (2.18) which is equivalent to (2.19).



Feldbaum (1961a,b, 1965) first recognized that for stochastic dynamic systems, the control action has a dual effect: It has direct affect on the state of the system but it also affects the uncertainty of the state. If the state-vector is augmented with the parameters, this observation by Feldbaum implies that the control action may be used to reduce the parameter uncertain and thus the controller's knowledge about the system. Due to the multi stage nature of the stochastic optimal control problem, it may be optimal for the controller to perturb the system to affect the states such that current system performance is degraded because these perturbations will lead to reduced parameter uncertainty and therefore improved future systems performance. This effect is called probing.

The existence of uncertainty in the system may have another effect. In general the existence of uncertainty in the system will increase the expected performance cost. The controller should therefore be cautious not to increase further the effect of the existing uncertainties on the cost. Compared to a deterministic controller, this will make the stochastic optimal controller cautious and less aggressive.

Bar-Shalom (1981) showed that the expected cost of the stochastic optimal control problem can be decomposed into a a term due to deterministic effect, a term related to the effect of exogenous stochastic variables, and a terms due to the uncertainties about the states (including uncertain parameters). When the effect of uncertainties dominate this cost one can distinguish two cases. In one case, the term related to the effect of exogenous variables dominate the expected cost. This situation corresponds to a highly uncertain model which cannot be improved during the control period. There is not much one can do to improve the control except being cautious and not being too confident in the model predictions. In the other uncertain case, when the dominating term in the expected objective function is due to the effect of uncertainties in the states and parameters, the controller may employ the dual action of control to reduce the uncertainties about the augmented state and thereby increase the future performance of the system. When the dominating term of the expected cost is related to deterministic effects, the controller can be of the certainty-equivalence type as it is approximately a deterministic problem.

As the uncertain stochastic dynamic system (2.18) may be interpreted as the stochastic dynamic system (2.19), we may apply the suboptimal controller discussed previously to the system (2.18) as well. These type of controllers are adaptive as they estimate the states of the system as well as the parameters in the model. The adaptive certainty equivalent controller applied to a stochastic optimal control problem with uncertain parameters may be stated as

**Algorithm 4 (Adaptive CEF Controller)**

The control input  $u_k = \tilde{\mu}_k(\mathcal{I}_k)$  applied by the certainty equivalent controller at each time  $k$  is determined by

1. Given the information vector  $\mathcal{I}_k$ , compute  $\hat{x}_{k|k} = E\{\mathbf{x}_k|\mathcal{I}_k\}$  and  $\hat{\theta}_k = E\{\theta|\mathcal{I}_k\}$

2. Solve the deterministic optimal control problem

$$\min \tilde{\phi} = \sum_{i=k}^{N-1} f_i(x_i, u_i, \bar{w}_i) + f_N(x_N) \quad (2.20a)$$

$$s.t. \quad x_k = \hat{x}_{k|k} \quad (2.20b)$$

$$x_{i+1} = g_i(x_i, u_i, \bar{w}_i, \hat{\theta}_k) \quad i = k, \dots, N-1 \quad (2.20c)$$

$$u_i \in U_i(x_i) \quad (2.20d)$$

for  $\{\tilde{u}_k, \tilde{u}_{k+1}, \dots, \tilde{u}_{N-1}\}$  using  $\bar{w}_i = E\{\mathbf{w}_i\}$ .

3. Apply the control input:  $\tilde{\mu}_k(\mathcal{I}_k) = \tilde{u}_k$ .

Seborg *et al.* (1986) as well as Ydstie (1997) have reviewed the application of adaptive certainty equivalent control to chemical processes. The solution of (2.20) is obtained as the solution of a standard optimal control problem for fixed parameters. The remaining issue in applying the adaptive certainty equivalent feedback controller concerns computation of the initial state estimate  $\hat{x}_{k|k}$  and the parameter estimate  $\hat{\theta}_k$ . In the algorithm these estimates are stated as the conditional means. However just as in the state estimation problem these estimates are more conveniently computed as either the maximum-a-posteriori (MAP) estimate (c.f. Melgaard, 1994; Seber and Wild, 1989) or as the least-squares estimate. The least squares estimate is obtained as the solution to the optimization problem

$$\min_{\theta, x_i, u_i, w_i} \psi \quad (2.21a)$$

$$s.t. \quad x_{i+1} = g_i(x_i, u_i, w_i, \theta) \quad i = 0, \dots, k-1 \quad (2.21b)$$

$$y_i = \varphi_i(x_i) + v_i \quad i = 0, \dots, k \quad (2.21c)$$

in which the objective function  $\psi$  is computed by

$$\begin{aligned} \psi = & \frac{1}{2}(\theta - \bar{\theta})' \Pi^{-1}(\theta - \bar{\theta}) \\ & + \frac{1}{2}(x_0 - \bar{x}_0)' P_0^{-1}(x_0 - \bar{x}_0) \\ & + \frac{1}{2} \sum_{i=0}^{k-1} (w_i - \bar{w}_i)' Q_i^{-1}(w_i - \bar{w}_i) \\ & + \frac{1}{2} \sum_{i=0}^k (v_i - \bar{v}_i)' R_i^{-1}(v_i - \bar{v}_i) \end{aligned} \quad (2.22)$$

The formulation (2.21) is easily extended with bounds on the states,  $x_k$ , estimated process noise,  $w_k$ , and the parameters,  $\theta$ , while bounds on the measurement noise  $v_k$  cannot be recommended. The mathematical program (2.21) can be solved by a general sparse solver for nonlinear programs. However, the structure as an optimal control problem is lost as the parameters  $\theta$  affect the state dynamics at every time step. Therefore the state vector is augmented

with the parameters such that the dynamics of the augmented system may be represented as

$$x_{k+1} = g_k(x_k, u_k, w_k, \theta_k) \quad (2.23a)$$

$$\theta_{k+1} = \theta_k \quad (2.23b)$$

Consequently, the least squares estimator (2.21) may be expressed as

$$\min_{z_i, u_i, w_i} \psi \quad (2.24a)$$

$$s.t. \quad z_{i+1} = \tilde{g}_i(z_i, u_i, w_i) \quad i = 0, \dots, k-1 \quad (2.24b)$$

$$y_i = \tilde{\varphi}_i(z_i) + v_i \quad i = 0, \dots, k \quad (2.24c)$$

in which the objective function is

$$\begin{aligned} \psi = & \frac{1}{2}(z_0 - \bar{z}_0)' \tilde{P}_0^{-1}(z_0 - \bar{z}_0) \\ & + \frac{1}{2} \sum_{i=0}^{k-1} (w_i - \bar{w}_i)' Q_i^{-1}(w_i - \bar{w}_i) \\ & + \frac{1}{2} \sum_{i=0}^k (v_i - \bar{v}_i)' R_i^{-1}(v_i - \bar{v}_i) \end{aligned} \quad (2.25)$$

and the augmented state  $z_k$  and other quantities used are

$$z_i = \begin{bmatrix} x_i \\ \theta_i \end{bmatrix} \quad \bar{z}_0 = \begin{bmatrix} \bar{x}_0 \\ \bar{\theta}_0 \end{bmatrix} \quad (2.26a)$$

$$\tilde{P}_0 = \begin{bmatrix} P_0 & 0 \\ 0 & \Pi \end{bmatrix} \quad (2.26b)$$

$$\tilde{g}_i(z_i, u_i, w_i) = \begin{bmatrix} g_i(x_i, u_i, w_i, \theta_i) \\ \theta_i \end{bmatrix} \quad (2.26c)$$

$$\tilde{\varphi}_i(z_i) = \varphi_i(x_i) \quad (2.26d)$$

It is immediately apparent that (2.24) has the same structure as (2.16). These problems do have the structure of an optimal control problem and may as a consequence be solved very efficiently by algorithms specialized to optimal control problems.

Using the state augmented representation (2.19) of the uncertain system (2.18) it may be regarded as the stochastic dynamic system discussed previously even though some of the states are parameters. Hence, in addition to the certainty equivalent feedback controller any of the other controllers for suboptimal solution of the stochastic optimal control problem may be applied to control of the stochastic dynamic system with uncertain parameters. The open-loop-optimal feedback controller for control of the uncertain system is called an adaptive open-loop-optimal feedback (OLOF) controller as it re-estimates (or adapt)

the parameters as well as the initial state as new information becomes available to the controller. The adaptive open-loop-optimal feedback controller may be stated as

**Algorithm 5 (Adaptive OLOF Controller)**

The control input  $u_k = \bar{\mu}_k(\mathcal{I}_k)$  applied by the adaptive open-loop-optimal feedback controller at each time  $k$  is determined by

1. Given the information vector  $\mathcal{I}_k$ , compute the conditional probability distributions  $P(x_k|\mathcal{I}_k)$  and  $P(\theta|\mathcal{I}_k)$ .
2. Let

$$\bar{\phi} = E_{x_k, \theta, \{w_i\}_{i=k, \dots, N-1}} \left\{ \sum_{i=k}^{N-1} f_i(\mathbf{x}_i, u_i, \mathbf{w}_i) + f_N(\mathbf{x}_N) \middle| \mathcal{I}_k \right\} \quad (2.27)$$

Compute a control sequence  $\{\bar{u}_k, \bar{u}_{k+1}, \dots, \bar{u}_{N-1}\}$  that solves

$$\min \bar{\phi} \quad (2.28a)$$

$$s.t. \quad \mathbf{x}_{i+1} = g_i(\mathbf{x}_i, u_i, \mathbf{w}_i, \theta), i = k, \dots, N-1 \quad (2.28b)$$

$$u_i \in U_i(\mathbf{x}_i), \quad i = k, \dots, N-1 \quad (2.28c)$$

3. Apply the control input

$$\bar{\mu}_k(\mathcal{I}_k) = \bar{u}_k \quad (2.29)$$

The conditional probability distributions in step 1 of the adaptive open-loop-optimal feedback controller may be computed approximately to the second order by computation of the conditional mean values and the conditional covariances. In practice the conditional mean values and the conditional covariances may be obtained as the optimal estimate and the associated sensitivities, respectively, of a least-squares problem. They may also be computed by solution of an appropriate maximum-a-posteriori (MAP) problem.

As previously discussed the open-loop-optimal feedback controller may suffer from the turn-off effect. The open-loop-optimal feedback controller accounts for future uncertainties but does not take future information into account. Therefore, it may be too cautious compared to the stochastic optimal controller.

## 2.3 Linear Quadratic Optimal Control

The model predictive controller described has been motivated and justified as being a mathematically tractable suboptimal controller for non-trivial processes. Yet, the requirement of on-line solution of the mathematical programs constituting the estimator and the regulator, respectively, limits the size of processes to which such a controller can be applied. To reduce this barrier, we describe in this chapter how the computationally intensive part of mathematical programs constituting the regulator and the estimator may be expressed as a structured constrained quadratic program for which highly efficient solution methods exist (c.f. Jørgensen *et al.*, 2004).

For notational convenience, let  $\mathcal{N} = \{0, 1, \dots, N-1\}$  and consider the discrete-time constrained linear quadratic optimal control problem of the Bolza type (c.f. Betts, 2001) represented by the quadratic program

$$\min_{\{x_{k+1}, u_k\}_{k \in \mathcal{N}}} \phi = \sum_{k \in \mathcal{N}} l_k(x_k, u_k) + l_N(x_N) \quad (2.30a)$$

$$s.t. \quad x_{k+1} = A'_k x_k + B'_k u_k + b_k, \quad k \in \mathcal{N} \quad (2.30b)$$

$$C'_k x_k + D'_k u_k + c_k \geq d_k, \quad k \in \mathcal{N} \quad (2.30c)$$

$$C'_N x_N + c_N \geq d_N \quad (2.30d)$$

in which  $x_0$  is a parameter and the stage costs are given by

$$l_k(x_k, u_k) = \frac{1}{2}(x'_k Q_k x_k + 2x'_k M_k u_k + u'_k R_k u_k) + q'_k x_k + r'_k u_k + f_k \quad k \in \mathcal{N} \quad (2.31a)$$

$$l_N(x_N) = \frac{1}{2}x'_N P_N x_N + p'_N x_N + \gamma_N \quad (2.31b)$$

For convex problems, the matrices  $R_0$ ,  $\begin{bmatrix} Q_k & M_k \\ M'_k & R_k \end{bmatrix}$  for  $k = 1, 2, \dots, N-1$ , and  $P_N$  are symmetric positive semi-definite matrices. Furthermore, a sufficient condition for the linear-quadratic optimal control problem to be strictly convex is that the matrices  $\{R_k\}_{k=0}^{N-1}$  are positive definite. However, this restriction is not necessary but merely sufficient for strict convexity of the linear-quadratic optimal control problem.

The corresponding unconstrained linear quadratic optimal control problem is

$$\min_{\{x_{k+1}, u_k\}_{k \in \mathcal{N}}} \phi = \sum_{k \in \mathcal{N}} l_k(x_k, u_k) + l_N(x_N) \quad (2.32a)$$

$$s.t. \quad x_{k+1} = A'_k x_k + B'_k u_k + b_k, \quad k \in \mathcal{N} \quad (2.32b)$$

It is important to note that the state transition equation (2.32b) is affine as it has the term  $b_k$ . The stage costs are general quadratic functions with quadratic terms as well as linear and zero order terms. Compared to the classic unconstrained linear quadratic control problem as it is typically described in textbooks, the formulation (2.32) with the stage costs (2.31) has an affine term  $b_k$  in the state transition equation and linear terms in the stage costs, i.e.  $q'_k x_k$ ,  $r'_k u_k$ , and  $p'_N x_N$ , as well as inconsequential zero order terms in the stage costs. The affine term in (2.32b) and the linear terms in (2.31) are important and arise naturally when the quadratic program is constructed from an unconstrained nonlinear optimal control problem as in sequential quadratic programming (c.f. Wright, 1993; Allgöwer *et al.*, 1999). Similarly, in the constrained case these terms as well as the affine terms in (2.30c)-(2.30d) are necessary in construction of the quadratic program used to compute the search direction in solution of the constrained nonlinear optimal control problem using an SQP algorithm (see appendix A).

The traditional formulation of the linear quadratic optimal control problem by control engineers has been in terms of deviation variables and with only quadratic terms in the objective function. The formulation in deviation variables implies that the affine term in (2.30b) vanishes. Similarly, for feasible set points the linear terms arising in the objective function from a least squares objective vanishes due the deviation variable formulation. Even when the setpoint is unreachable the linear terms of the unconstrained optimal control problem are inconsequential as they vanish for the optimal solution due orthogonality in Hilbert spaces between the optimal solution and its corresponding setpoint-error (c.f. Luenberger, 1969). However, expression of the classic LQ tracking or disturbance rejection problem as well as extensions such as the anticipatory control problem are facilitated by the formulation including linear terms in the stage costs and affine terms in the state transition equation. In the constrained case, the affine and linear terms may become indispensable in situations where the setpoint is infeasible and inequality constraints are active.

### 2.3.1 Numerical Methods

Jørgensen *et al.* (2004) provide an overview of efficient solution method for the linear quadratic optimal control problem. These methods has been tailored such that they utilize the structure of the linear-quadratic optimal control problem. This solution procedure factorizes the arising KKT-matrices using Riccati iteration. In an interior-point context, Steinbach (1994), Rao *et al.* (1998), and Tenny and Rawlings (2002) considered solution of the linear-quadratic optimal control problem by Riccati iteration. Bartlett *et al.* (2000) compared interior-point methods and active set methods for nonlinear model predictive control. They did not use Riccati iteration solver but applied general sparse as well as dense solvers for solution of the KKT-systems associated with the linear-quadratic optimal control problem. Diehl (2001) mentions dense as well as interior-point algorithms for solution of the quadratic programs arising in algorithms for nonlinear model predictive control.

The purpose of this section is merely to outline the major computational operations used in solving constrained linear-quadratic optimal control problems. This serves to motivate specially tailored algorithms for constrained linear-quadratic optimal control problems. As outlined in chapter 3, the constrained linear-quadratic optimal control problem (2.30) may be solved efficiently solving a sequence of unconstrained linear-quadratic optimal control problems (2.32). The Lagrangian of (2.32) may be expressed as

$$\begin{aligned} \mathcal{L}(x, u, \pi) = & \sum_{k \in \mathcal{N}} l_k(x_k, u_k) + l_N(x_N) \\ & - \sum_{k \in \mathcal{N}} \pi'_k (x_{k+1} - (A'_k x_k + B'_k u_k + b_k)) \end{aligned} \quad (2.33)$$

and the corresponding first order Karush-Kuhn-Tucker conditions are

$$Q_k x_k + M_k u_k + q_k - \pi_{k-1} + A_k \pi_k = 0 \quad k \in \mathcal{N} \setminus \{0\} \quad (2.34a)$$

$$M'_k x_k + R_k u_k + r_k + B_k \pi_k = 0 \quad k \in \mathcal{N} \quad (2.34b)$$

$$P_N x_N + p_N - \pi_{N-1} = 0 \quad (2.34c)$$

$$x_{k+1} = A'_k x_k + B'_k u_k + b_k \quad k \in \mathcal{N} \quad (2.34d)$$

For the case  $N = 3$ , the KKT conditions (2.34) may be stated as the following symmetric structured linear system of equations

$$\left[ \begin{array}{cccc|cccc} R_0 & & & & B_0 & & & \\ & Q_1 & M_1 & & -I & A_1 & & \\ & M'_1 & R_1 & & & B_1 & & \\ & & & Q_2 & M_2 & & & \\ & & & M'_2 & R_2 & & & \\ & & & & & & P_3 & \\ \hline B'_0 & -I & & & & & & \\ & A'_1 & B'_1 & -I & & & & \\ & & & A'_2 & B'_2 & -I & & \end{array} \right] \begin{bmatrix} u_0 \\ x_1 \\ u_1 \\ x_2 \\ u_2 \\ x_3 \\ \pi_0 \\ \pi_1 \\ \pi_2 \end{bmatrix} = - \begin{bmatrix} r_0 + M'_0 x_0 \\ q_1 \\ r_1 \\ q_2 \\ r_2 \\ p_3 \\ b_0 + A'_0 x_0 \\ b_1 \\ b_2 \end{bmatrix} \quad (2.35)$$

which may be rearranged to the symmetric block-diagonal system of equations

$$\left[ \begin{array}{cccccccc} R_0 & B_0 & & & & & & \\ B'_0 & 0 & -I & & & & & \\ & & & Q_1 & M_1 & A_1 & & \\ & & & M'_1 & R_1 & B_1 & & \\ & & & A'_1 & B'_1 & 0 & -I & \\ & & & & & -I & Q_2 & M_2 & A_2 \\ & & & & & & M'_2 & R_2 & B_2 \\ & & & & & & A'_2 & B'_2 & 0 & -I \\ & & & & & & & & -I & P_3 \end{array} \right] \begin{bmatrix} u_0 \\ \pi_0 \\ x_1 \\ u_1 \\ \pi_1 \\ x_2 \\ u_2 \\ \pi_2 \\ x_3 \end{bmatrix} = - \begin{bmatrix} r_0 + M'_0 x_0 \\ b_0 + A'_0 x_0 \\ q_1 \\ r_1 \\ b_1 \\ q_2 \\ r_2 \\ b_2 \\ p_3 \end{bmatrix} \quad (2.36)$$

The solution of the KKT-system (2.36) may be accomplished by a general sparse symmetric solver (Biegler, 2000), by a block-diagonal solver (Wright, 1996), or by a Riccati iteration based solver. However, general dense solvers are unsuitable for solution of systems of the type (2.36) as they easily become large for moderate to large values of the horizon,  $N$ . The Riccati iteration based solver is a solution procedure particularly developed for solution of the unconstrained linear-quadratic optimal control problem (2.32). As a consequence it is also very efficient for solution of linear systems of the type (2.36). The Riccati based factorization procedure for solution of the unconstrained linear-quadratic optimal control problem and system of the type (2.36) is stated in the following proposition.

**Proposition 2.3.1 (Optimal Solution)**

The optimal primal-dual solution,  $\{x_{k+1}, u_k, \pi_k\}_{k=0}^{N-1}$ , of the linear-quadratic optimal control, (2.32) with the stage costs (2.31), may be obtained by the following procedure

1. Compute

$$K_k = -(R_k + B_k P_{k+1} B'_k)^{-1} (M'_k + B_k P_{k+1} A'_k) \quad (2.37a)$$

$$a_k = -(R_k + B_k P_{k+1} B'_k)^{-1} (r_k + B_k (P_{k+1} b_k + p_{k+1})) \quad (2.37b)$$

$$P_k = Q_k + A_k P_{k+1} A'_k + (M_k + A_k P_{k+1} B'_k) K_k \quad (2.37c)$$

$$p_k = (A_k + K'_k B_k) (P_{k+1} b_k + p_{k+1}) + q_k + K'_k r_k \quad (2.37d)$$

for  $k = N-1, N-2, \dots, 0$ .

2. Compute the primal solution  $\{u_k, x_{k+1}\}_{k=0}^{N-1}$  for  $k = 0, 1, \dots, N-1$  by

$$u_k = K_k x_k + a_k \quad (2.38a)$$

$$x_{k+1} = A'_k x_k + B'_k u_k + b_k \quad (2.38b)$$

3. Obtain the dual solution  $\{\pi_k\}_{k=0}^{N-1}$  by computing

$$\pi_{N-1} = P_N x_N + p_N \quad (2.39a)$$

$$\pi_{k-1} = A_k \pi_k + Q_k x_k + M_k u_k + q_k \quad (2.39b)$$

for  $k = N-1, N-2, \dots, 1$ .

*Proof.* See chapter 4. □

### Remark 2.3.2

$\{K_k, P_k\}_{k=0}^{N-1}$  may be regarded as a factorization of the KKT matrix of (2.36). This factorization does not change as long as the matrices in  $\{A_k, B_k, Q_k, M_k, R_k\}_{k=0}^{N-1} \cup \{P_N\}$  do not change.

### Remark 2.3.3

The computational complexity of the Riccati iteration based solution procedure stated in proposition 2.3.1 is  $O(Nm^3)$ . Alternative solution procedures based on elimination of the states has either computational complexity  $O(N^3m^3)$  or  $O(N^2m^2)$  depending on the method of implementation. For long prediction horizons,  $N$ , the Riccati based solution procedure is at least one order of magnitude faster than alternative solution procedure based on solution of dense quadratic programs.

Along with the computation of the optimal solution by proposition 2.3.1, the optimal cost-to-go at each stage may be computed cheaply. The basis for this computation is stated in the following proposition.

### Proposition 2.3.4 (Optimal Value)

The optimal value  $\phi^* = V(x_0)$  of the unconstrained linear-quadratic optimal control problem, (2.32) with stage costs (2.31), is a function of the initial stage given by

$$\phi^* = V(x_0) = \frac{1}{2} x'_0 P_0 x_0 + p'_0 x_0 + \gamma_0 \quad (2.40)$$

in which  $P_0$  and  $p_0$  are computed by the recursion (2.37) and  $\gamma_0$  is computed by the recursion

$$\begin{aligned} \gamma_k &= \gamma_{k+1} + f_k + p'_{k+1} b_k + \frac{1}{2} b'_k P_{k+1} b_k \\ &\quad - \frac{1}{2} (r_k + B_k (P_{k+1} b_k + p_{k+1}))' (R_k + B_k P_{k+1} B'_k)^{-1} (r_k + B_k (P_{k+1} b_k + p_{k+1})) \end{aligned} \quad (2.41)$$



for  $k = N - 1, N - 2, \dots, 1, 0$ .

*Proof.* See chapter 4. □

**Remark 2.3.5**

The optimal cost-to-go at each stage is

$$V_k(x_k) = \frac{1}{2}x_k'P_kx_k + p_k'x_k + \gamma_k \quad (2.42)$$

**Remark 2.3.6**

By proposition 2.3.4, the optimal value  $\phi^* = V_0(x_0) = V(x_0) = \frac{1}{2}x_0'P_0x_0 + p_0'x_0 + \gamma_0$  of the unconstrained linear-quadratic optimal control problem may be computed in advance of the optimal solution  $\{x_{k+1}^*, u_k^*, \pi_k^*\}_{k=0}^{N-1}$ . This implies that we may know the value of the unconstrained linear-quadratic optimal control problem without knowing its solution.

The constrained linear-quadratic optimal control problem (2.30) may also be solved by elimination of the states using the state transition equations. The resulting problem is a dense quadratic program with the control inputs,  $\{u_k\}_{k=0}^{N-1}$ , as decision variables. This quadratic program may be solved by either a primal active set quadratic program algorithm (c.f. Gill and Murray, 1978; Gill *et al.*, 1991, 1995), a dual active set quadratic program algorithm (c.f. Goldfarb and Idnani, 1983; Schmid and Biegler, 1994; Bartlett *et al.*, 2000), or an interior-point algorithm for convex quadratic programs (c.f. Mehrotra, 1992, 1993; Wright, 1997; Nocedal and Wright, 1999).

## 2.4 Nonlinear Optimal Control

Consider the discrete-time finite horizon constrained nonlinear optimal control problem

$$\min_{\{x_{k+1}, u_k\}_{k \in \mathcal{N}}} \phi = \sum_{k \in \mathcal{N}} f_k(x_k, u_k) + f_N(x_N) \quad (2.43a)$$

$$s.t. \quad x_{k+1} = g_k(x_k, u_k) \quad k \in \mathcal{N} \quad (2.43b)$$

$$h_k(x_k, u_k) \geq d_k \quad k \in \mathcal{N} \quad (2.43c)$$

$$h_N(x_N) \geq d_N \quad (2.43d)$$

in which the initial state  $x_0 \in \mathbb{R}^n$  is assumed specified and  $\mathcal{N} = \{0, 1, \dots, N - 1\}$ . The functions  $f_k : \mathbb{R}^n \times \mathbb{R}^m \mapsto \mathbb{R}$  for  $k \in \mathcal{N}$ ,  $f_N : \mathbb{R}^n \mapsto \mathbb{R}$ ,  $g_k : \mathbb{R}^n \times \mathbb{R}^m \mapsto \mathbb{R}^n$  for  $k \in \mathcal{N}$ ,  $h_k : \mathbb{R}^n \times \mathbb{R}^m \mapsto \mathbb{R}^{p_k}$  for  $k \in \mathcal{N}$ , and  $h_N : \mathbb{R}^n \mapsto \mathbb{R}^{p_N}$  are assumed to be sufficiently smooth. This problem arises, for instance in the discretization of a corresponding continuous time optimal control problem (c.f. Betts, 2001; Martinsen *et al.*, 2002). With this origin, the involved functions are the combined result of the particular discretization applied and the underlying continuous time physics of the system.

As is evident by the discussion in appendix A, the sequential quadratic programming algorithm solves nonlinear programs by computing a sequence of search directions as the solution of some quadratic programs. The constraints of these quadratic programs are linearizations of the constraints in the nonlinear program around the current iterate. The gradient of the objective function in the quadratic program is the gradient of the objective function in the non-linear program at the current iterate. The Hessian matrix in the quadratic program is some approximation of the Hessian of the Lagrangian of the nonlinear program at the current iterate.

To construct the quadratic programs used in an SQP algorithm for solution of (2.43), the constraints (2.43b)-(2.43d) must be linearized. This linearization is stated in the following lemma.

**Lemma 2.4.1**

Let  $x_0$  be given. Let  $\{x_{k+1}^0, u_k^0\}_{k=0}^{N-1}$  be some given nominal trajectory of the nonlinear optimal control problem (2.43).

Then the equations

$$\Delta x_{k+1} = A'_k \Delta x_k + B'_k \Delta u_k + b_k \quad k \in \mathcal{N} \quad (2.44a)$$

$$C'_k \Delta x_k + D'_k \Delta u_k + c_k \geq d_k \quad k \in \mathcal{N} \quad (2.44b)$$

$$C'_N \Delta x_N + c_N \geq d_N \quad (2.44c)$$

are a linearization of the constraints (2.43b)-(2.43d) in the nonlinear optimal control problem (2.43). The initial state,  $\Delta x_0$ , in (2.44a) is

$$\Delta x_0 = 0 \quad (2.45)$$

and the matrices at time index  $k = 0$  are

$$A_0 = \nabla_{x_0} g_0(x_0, u_0^0) \quad C_0 = \nabla_{x_0} h_0(x_0, u_0^0) \quad (2.46a)$$

$$B_0 = \nabla_{u_0} g_0(x_0, u_0^0) \quad D_0 = \nabla_{u_0} h_0(x_0, u_0^0) \quad (2.46b)$$

$$b_0 = g_0(x_0, u_0^0) - x_1^0 \quad c_0 = h_0(x_0, u_0^0) \quad (2.46c)$$

The matrices at time  $k = 1, 2, \dots, N-1$  are

$$A_k = \nabla_{x_k} g_k(x_k^0, u_k^0) \quad C_k = \nabla_{x_k} h_k(x_k^0, u_k^0) \quad (2.47a)$$

$$B_k = \nabla_{u_k} g_k(x_k^0, u_k^0) \quad D_k = \nabla_{u_k} h_k(x_k^0, u_k^0) \quad (2.47b)$$

$$b_k = g_k(x_k^0, u_k^0) - x_{k+1}^0 \quad c_k = h_k(x_k^0, u_k^0) \quad (2.47c)$$

and the matrices associated with the final time  $k = N$  are

$$C_N = \nabla_{x_N} h_N(x_N^0) \quad c_N = h_N(x_N^0) \quad (2.48)$$

*Proof.* Follows directly by linearization of (2.43b)-(2.43d) around  $\{x_{k+1}^0, u_k^0\}_{k=0}^{N-1}$  and with  $x_0$  specified as a parameter.  $\square$

**Remark 2.4.2**

In equation (2.46a),  $A_0$  and  $C_0$  are defined as the derivatives of  $g$  and  $h$ , respectively, with respect to  $x_0$ . As  $\Delta x_0 = 0$ , the matrices  $A_0$  and  $C_0$  have no influence on the state dynamics. Therefore, they are not used in actual computations and do not need to be evaluated. Alternatively, they could be defined as  $A_0 = 0$  and  $C_0 = 0$ . They are included in the formulation of (2.44) such that this formulation is compatible with the equations (2.43b)-(2.43d).

As discussed in appendix A, the objective function

$$\psi(\eta) = \frac{1}{2}\eta'W\eta + t'\eta \quad (2.49)$$

of the quadratic program used in an SQP algorithm, is constructed such that coefficients of the linear term,  $t$ , is the gradient of the nonlinear objective function evaluated at the current iterate. The Hessian matrix,  $W$ , is selected as some approximation of the Hessian of the Lagrangian with respect to the primal variables.

The Lagrangian,  $\mathcal{L}$ , of the nonlinear optimal control problem (2.43) is

$$\begin{aligned} \mathcal{L} = & \sum_{k \in \mathcal{N}} f_k(x_k, u_k) + f_N(x_N) \\ & - \sum_{k \in \mathcal{N}} \pi'_k(x_{k+1} - g_k(x_k, u_k)) \\ & - \sum_{k \in \mathcal{N}} \lambda'_k(h_k(x_k, u_k) - d_k) \\ & - \lambda'_N(h_N(x_N) - d_N) \end{aligned} \quad (2.50)$$

The gradients of the Lagrangian with respect to the primal variables are

$$\begin{aligned} \nabla_{x_k} \mathcal{L} = & \nabla_{x_k} f_k(x_k, u_k) - \pi_{k-1} \\ & + \nabla_{x_k} g_k(x_k, u_k) \pi_k - \nabla_{x_k} h_k(x_k, u_k) \lambda_k \end{aligned} \quad (2.51a)$$

$$\nabla_{x_N} \mathcal{L} = \nabla_{x_N} f_N(x_N) - \pi_{N-1} - \nabla_{x_N} h_N(x_N) \lambda_N \quad (2.51b)$$

$$\begin{aligned} \nabla_{u_k} \mathcal{L} = & \nabla_{u_k} f_k(x_k, u_k) + \nabla_{u_k} g_k(x_k, u_k) \pi_k \\ & - \nabla_{u_k} h_k(x_k, u_k) \lambda_k \end{aligned} \quad (2.51c)$$

From these expressions, it is evident that the gradients are separable with respect to the primal variables,  $\{x_{k+1}, u_k\}_{k=0}^{N-1}$ . They are separable in the sense that the gradients with respect to the primal variables  $(x_k, u_k)$  depend on the primal variables  $(x_k, u_k)$  associated with the time index  $k$ , but not on the primal variables  $(x_j, u_j)$  associated with other time indices, i.e.  $j \neq k$ . Consequently, the Hessian matrix of the Lagrangian with respect to the primal variables,  $\{x_{k+1}, u_k\}_{k=0}^{N-1}$ , is block-diagonal, as the Lagrangian gradients (2.51)

are separable with respect to the primal variables. The diagonal blocks are

$$\begin{aligned}\nabla_{x_k, x_k}^2 \mathcal{L} &= \nabla_{x_k, x_k}^2 f_k(x_k, u_k) \\ &\quad - \sum_{i=1}^n (\pi_k)_i \nabla_{x_k, x_k}^2 (g_k)_i(x_k, u_k) \\ &\quad - \sum_{i=1}^p (\lambda_k)_i \nabla_{x_k, x_k}^2 (h_k)_i(x_k, u_k)\end{aligned}\tag{2.52a}$$

$$\begin{aligned}\nabla_{x_k, u_k}^2 \mathcal{L} &= \nabla_{x_k, u_k}^2 f_k(x_k, u_k) \\ &\quad - \sum_{i=1}^n (\pi_k)_i \nabla_{x_k, u_k}^2 (g_k)_i(x_k, u_k) \\ &\quad - \sum_{i=1}^p (\lambda_k)_i \nabla_{x_k, u_k}^2 (h_k)_i(x_k, u_k)\end{aligned}\tag{2.52b}$$

$$\begin{aligned}\nabla_{u_k, u_k}^2 \mathcal{L} &= \nabla_{u_k, u_k}^2 f_k(x_k, u_k) \\ &\quad - \sum_{i=1}^n (\pi_k)_i \nabla_{u_k, u_k}^2 (g_k)_i(x_k, u_k) \\ &\quad - \sum_{i=1}^p (\lambda_k)_i \nabla_{u_k, u_k}^2 (h_k)_i(x_k, u_k)\end{aligned}\tag{2.52c}$$

at times  $k = 0, 1, \dots, N-1$  and

$$\begin{aligned}\nabla_{x_N, x_N}^2 \mathcal{L} &= \nabla_{x_N, x_N}^2 f_N(x_N) \\ &\quad - \sum_{i=1}^p (\lambda_N)_i \nabla_{x_N, x_N}^2 (h_N)_i(x_N)\end{aligned}\tag{2.52d}$$

at the final time  $k = N$ . Let

$$w_k = \begin{bmatrix} x_k \\ u_k \end{bmatrix} \quad k = 0, 1, \dots, N-1\tag{2.53a}$$

$$w_N = x_N\tag{2.53b}$$

and

$$w = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_N \end{bmatrix} \quad \pi = \begin{bmatrix} \pi_0 \\ \pi_1 \\ \vdots \\ \pi_{N-1} \end{bmatrix} \quad \lambda = \begin{bmatrix} \lambda_0 \\ \lambda_1 \\ \vdots \\ \lambda_N \end{bmatrix}\tag{2.54}$$

Then the objective function  $\psi = \psi(\Delta w)$  of the quadratic program in the SQP algorithm for the nonlinear optimal control problem becomes

$$\begin{aligned}\psi &= \frac{1}{2} \Delta w' \nabla_{ww}^2 \mathcal{L}(w, \pi, \lambda) \Delta w + \nabla \phi(w)' \Delta w \\ &= \sum_{k=0}^N \frac{1}{2} \Delta w_k' \nabla_{w_k w_k}^2 \mathcal{L}(w, \pi, \lambda) \Delta w_k + \nabla_{w_k} \phi(w)' \Delta w_k\end{aligned}\tag{2.55}$$

in which Hessian matrix  $W$  of the quadratic program is chosen to be the Hessian of the Lagrangian, i.e.  $W = \nabla_{ww}^2 \mathcal{L}(w, \pi, \lambda)$ . The second equality in (2.55) is obtained using the block-diagonal structure of the Hessian matrix.

Lemma 2.4.3 states a quadratic approximation of the Lagrangian of the non-linear optimal control problem with respect to the states,  $\{x_k\}_{k=0}^N$ , and the control inputs,  $\{u_k\}_{k=0}^{N-1}$ . In conjunction with lemma 2.4.1, this result is used in proposition 2.4.4 to state the quadratic program used for computing the search direction in a sequential quadratic programming algorithm based on the exact Hessian of the Lagrangian. The resulting quadratic program is a constrained linear-quadratic optimal control problem.

**Lemma 2.4.3**

Let  $x_0$  be given. Let  $\{x_{k+1}^0, u_k^0\}_{k=0}^{N-1}$  be the trajectory corresponding to the current iterate of an SQP algorithm. Let  $\{\pi_k^0\}_{k=0}^{N-1}$  and  $\{\lambda_k^0\}_{k=0}^N$  be the associated trajectories of the dual variables of (2.43). Let the Hessian of the quadratic program in an SQP algorithm be equal to the Hessian of the Lagrangian with respect to the primal variables of (2.43).

Then the objective function of the quadratic program is

$$\psi = \sum_{k=0}^{N-1} l_k(\Delta x_k, \Delta u_k) + l_N(\Delta x_N) \quad (2.56)$$

in which

$$\begin{aligned} l_k(\Delta x_k, \Delta u_k) = & \frac{1}{2} \Delta x_k' Q_k \Delta x_k + \Delta x_k' M_k \Delta u_k \\ & + \frac{1}{2} \Delta u_k' R_k \Delta u_k + q_k' \Delta x_k \\ & + r_k' \Delta u_k + f_k \quad k \in \mathcal{N} \end{aligned} \quad (2.57a)$$

$$l_N(\Delta x_N) = \frac{1}{2} \Delta x_N' P_N \Delta x_N + p_N' \Delta x_N + \gamma_N \quad (2.57b)$$

and  $\Delta x_0 = 0$ . The parameters at  $k = 0$  are computed by

$$Q_0 = \nabla_{x_0, x_0}^2 \mathcal{L}(x_0, u_0^0, \pi_0^0, \lambda_0^0) \quad (2.58a)$$

$$M_0 = \nabla_{x_0, u_0}^2 \mathcal{L}(x_0, u_0^0, \pi_0^0, \lambda_0^0) \quad (2.58b)$$

$$R_0 = \nabla_{u_0, u_0}^2 \mathcal{L}(x_0, u_0^0, \pi_0^0, \lambda_0^0) \quad (2.58c)$$

$$q_0 = \nabla_{x_0} f_0(x_0, u_0^0) \quad (2.58d)$$

$$r_0 = \nabla_{u_0} f_0(x_0, u_0^0) \quad (2.58e)$$

$$f_0 = f_0(x_0, u_0^0) \quad (2.58f)$$

Similarly, the parameters at  $k = 1, 2, \dots, N-1$  are computed by

$$Q_k = \nabla_{x_k, x_k}^2 \mathcal{L}(x_k^0, u_k^0, \pi_k^0, \lambda_k^0) \quad (2.59a)$$

$$M_k = \nabla_{x_k, u_k}^2 \mathcal{L}(x_k^0, u_k^0, \pi_k^0, \lambda_k^0) \quad (2.59b)$$

$$R_k = \nabla_{u_k, u_k}^2 \mathcal{L}(x_k^0, u_k^0, \pi_k^0, \lambda_k^0) \quad (2.59c)$$

$$q_k = \nabla_{x_k} f_k(x_k^0, u_k^0) \quad (2.59d)$$

$$r_k = \nabla_{u_k} f_k(x_k^0, u_k^0) \quad (2.59e)$$

$$f_k = f_k(x_k^0, u_k^0) \quad (2.59f)$$

Finally, the parameters at  $k = N$  are

$$P_N = \nabla_{x_N, x_N}^2 \mathcal{L}(x_N^0, \lambda_N^0) \quad (2.60a)$$

$$p_N = \nabla_{x_N} f_N(x_N^0) \quad (2.60b)$$

$$\gamma_N = f_N(x_N^0) \quad (2.60c)$$

*Proof.* The result follows directly by a specialization of (2.55).  $\square$

#### Proposition 2.4.4

Let  $x_0$  be given. Let  $\{x_{k+1}^0, u_k^0\}_{k=0}^{N-1}$  be the trajectory corresponding to the current iterate of an SQP algorithm applied to (2.43). Let  $\{\pi_k^0\}_{k=0}^{N-1}$  and  $\{\lambda_k^0\}_{k=0}^N$  be the associated trajectories of the dual variables of (2.43). Let the Hessian of the quadratic program in an SQP algorithm be equal to the Hessian of the Lagrangian with respect to the primal variables,  $\{x_{k+1}, u_k\}_{k=0}^{N-1}$ , of (2.43).

Then the quadratic program for computation of the search direction in an SQP algorithm applied to (2.43) is

$$\min_{\{\Delta x_{k+1}, \Delta u_k\}} \psi = \sum_{k=0}^{N-1} l_k(\Delta x_k, \Delta u_k) + l_N(\Delta x_N) \quad (2.61a)$$

$$s.t. \quad \Delta x_{k+1} = A'_k \Delta x_k + B'_k \Delta u_k + b_k \quad (2.61b)$$

$$C'_k \Delta x_k + D'_k \Delta u_k + c_k \geq d_k \quad (2.61c)$$

$$C'_N \Delta x_N + c_N \geq d_N \quad (2.61d)$$

in which the objective function is computed according to lemma 2.4.3 and the constraints are computed according to lemma 2.4.1. The initial state is  $\Delta x_0 = 0$ .

*Proof.* The result is valid for SQP algorithms that construct the quadratic programs using the exact Hessian of the Lagrangian function. Therefore, the result follows straightforwardly from lemma 2.4.1 and 2.4.3. The initial state in the constrained linear-quadratic optimal control problem is  $\Delta x_0 = 0$  as  $x_0$  is a parameter and not a decision variable in the nonlinear optimal control problem.  $\square$

The matrices  $\begin{bmatrix} Q_k & M_k \\ M'_k & R_k \end{bmatrix}$ ,  $P_N$ , and  $R_k$  in proposition 2.4.4 are obtained as the exact Hessian of the Lagrangian, i.e. as the secondary derivatives of the Lagrangian. These matrices are not necessarily positive definite or even positive semi-definite. They may be indefinite. Most current quadratic programming solvers are limited to convex or strictly convex quadratic programs. Hence, they cannot be applied for solving the constrained linear-quadratic optimal control problem stated in proposition 2.4.4. However, the primal active set quadratic programming algorithm described by Gill and Murray (1978) and Gill *et al.* (1991, 1995) is applicable to indefinite as well as convex quadratic programs.

### 2.4.1 BFGS Approximation

In the solution of general nonlinear programs with the SQP algorithm, quasi-Newton methods are employed when the exact Hessian computationally is too expensive to evaluate or a strictly positive definite Hessian is demanded due to the method used for solving the quadratic program (c.f. Nocedal and Wright, 1999; Dennis and Schnabel, 1996). The most popular quasi-Newton method employs the damped BFGS update of the Hessian matrix.

Let  $v$  be the vector of decision variables in the nonlinear optimal control problem (2.43)

$$v = \begin{bmatrix} u_0 \\ x_1 \\ u_1 \\ x_2 \\ \vdots \\ u_{N-1} \\ x_N \end{bmatrix} \quad (2.62)$$

Then as described in appendix A, the damped BFGS update is obtained by the recursion

$$W^{(i+1)} = W^{(i)} - \frac{W^{(i)} s s' W^{(i)}}{s' W^{(i)} s} + \frac{r r'}{s' r} \quad (2.63a)$$

in which

$$s = v^{(i+1)} - v^{(i)} \quad (2.63b)$$

$$y = \nabla_v \mathcal{L}(v^{(i+1)}, \pi, \lambda) - \nabla_v \mathcal{L}(v^{(i)}, \pi, \lambda) \quad (2.63c)$$

$$r = \theta y + (1 - \theta) W^{(i)} s \quad (2.63d)$$

and

$$\theta = \begin{cases} 1 & s' y \geq 0.2 s' W^{(i)} s \\ \frac{0.8 s' W^{(i)} s}{s' W^{(i)} s - s' y} & s' y < 0.2 s' W^{(i)} s \end{cases} \quad (2.63e)$$

The Hessian matrix,  $W^{(i)}$ , obtained at iteration  $i$  by this method is obviously dense. This implies that the objective function

$$\psi(\Delta v) = \frac{1}{2} \Delta v' W^{(i)} \Delta v + \nabla \phi(v)' \Delta v \quad (2.64)$$

is not partially separable as (2.55). Consequently, an approximation of the Hessian based on (2.63) will be computationally inefficient as it does not retain the structure of a linear quadratic optimal control problem and has all the states as well as the inputs as decision variables.

To overcome this computational deficiency of the standard damped BFGS update, Bock *et al.* (2000) suggested to modify the update to each block of decision variables such that the approximated Hessian has the same structure as

the Hessian computed by lemma 2.4.3. In the context of unconstrained optimization of partially separable functions, such an approach had previously been proposed by Griewank and Toint (1982). Rakshit and Sen (1990) considered the structured update of the Hessian matrix in the context of differential dynamic programming. The block diagonal Hessian matrix,  $W$ , is

$$W = \begin{bmatrix} W_0 & & & & \\ & W_1 & & & \\ & & \ddots & & \\ & & & W_{N-1} & \\ & & & & W_N \end{bmatrix} \quad (2.65)$$

The block-diagonal structure of the Hessian matrix,  $W$ , implies that the objective function of the quadratic program in an SQP algorithm becomes

$$\begin{aligned} \psi(\Delta v) &= \frac{1}{2} \Delta v' W \Delta v + \nabla \phi(v)' \Delta v \\ &= \sum_{k=0}^N \frac{1}{2} \Delta v_k' W_k \Delta v_k + \nabla_{v_k} \phi(v)' \Delta v_k \end{aligned} \quad (2.66)$$

in which

$$v_0 = u_0 \quad (2.67a)$$

$$v_k = \begin{bmatrix} x_k \\ u_k \end{bmatrix} \quad k = 1, 2, \dots, N-1 \quad (2.67b)$$

$$v_N = x_N \quad (2.67c)$$

The function in (2.66) has the same partially separable structure as the function in (2.55). The only difference is that the parameter  $x_0$  is left out in the specification of (2.66).

Let  $\mathcal{L}^{(i)} = \mathcal{L}(v^{(i)}, \pi, \lambda)$  and  $\mathcal{L}^{(i+1)} = \mathcal{L}(v^{(i+1)}, \pi, \lambda)$ , in which  $\pi = \pi^{(i+1)}$  and  $\lambda = \lambda^{(i+1)}$ . For  $k = 0, 1, \dots, N$ , each diagonal block  $W_k$  of the Hessian matrix  $W$  is approximated by the damped BFGS recursion

$$W_k^{(i+1)} = W_k^{(i)} - \frac{W_k^{(i)} s_k s_k' W_k^{(i)}}{s_k' W_k^{(i)} s_k} + \frac{r_k r_k'}{s_k' r_k} \quad (2.68a)$$

in which

$$r_k = \theta_k y_k + (1 - \theta_k) W_k^{(i)} s_k \quad (2.68b)$$

and

$$\theta_k = \begin{cases} 1 & s_k' y_k \geq 0.2 s_k' W_k^{(i)} s_k \\ \frac{0.8 s_k' W_k^{(i)} s_k}{s_k' W_k^{(i)} s_k - s_k' y_k} & s_k' y_k < 0.2 s_k' W_k^{(i)} s_k \end{cases} \quad (2.68c)$$



The vectors used for updating the Hessian block corresponding to time index  $k = 0$  are computed by

$$s_0 = u_0^{(i+1)} - u_0^{(i)} \quad (2.68d)$$

$$y_0 = \nabla_{u_0} \mathcal{L}^{(i+1)} - \nabla_{u_0} \mathcal{L}^{(i)} \quad (2.68e)$$

Similarly, the vectors used for updating the Hessian diagonal block corresponding to time indices  $k = 1, 2, \dots, N-1$  are computed by

$$s_k = \begin{bmatrix} x_k^{(i+1)} - x_k^{(i)} \\ u_k^{(i+1)} - u_k^{(i)} \end{bmatrix} \quad (2.68f)$$

$$y_k = \begin{bmatrix} \nabla_{x_k} \mathcal{L}^{(i+1)} - \nabla_{x_k} \mathcal{L}^{(i)} \\ \nabla_{u_k} \mathcal{L}^{(i+1)} - \nabla_{u_k} \mathcal{L}^{(i)} \end{bmatrix} \quad (2.68g)$$

By an analogous recipe, the vectors for computing the Hessian diagonal block corresponding to time index  $k = N$  are obtained by the equations

$$s_N = x_N^{(i+1)} - x_N^{(i)} \quad (2.68h)$$

$$y_N = \nabla_{x_N} \mathcal{L}^{(i+1)} - \nabla_{x_N} \mathcal{L}^{(i)} \quad (2.68i)$$

The just discussed procedure for computing a positive definite quadratic approximation of the Hessian of the Lagrangian around the current iterate may be applied for computing a quadratic approximation of the Lagrangian. This quadratic approximation is the objective function of the quadratic program used for computing the search direction in line-search SQP algorithms. The construction of the objective function in the quadratic program by the structured BFGS update is summarized in the following lemma.

**Lemma 2.4.5**

Let  $x_0$  be given. Let  $\{x_{k+1}^{(i+1)}, u_k^{(i+1)}\}_{k=0}^{N-1}$  and  $\{x_{k+1}^{(i)}, u_k^{(i)}\}_{k=0}^{N-1}$  be trajectories at two subsequent iterations of an SQP algorithm applied to (2.43). Let  $\{\pi_k\}_{k=0}^{N-1} = \{\pi_k^{(i+1)}\}_{k=0}^{N-1}$  and  $\{\lambda_k\}_{k=0}^N = \{\lambda_k^{(i+1)}\}_{k=0}^N$  be the associated trajectories of the dual variables of (2.43). Let the Hessian matrix of the quadratic program be computed by the structured damped BFGS procedure (2.68).

The the objective function of the quadratic program in the SQP algorithm for generation of the search direction  $\{\Delta x_{k+1}, \Delta u_k\}_{k=0}^{N-1} = \{\Delta x_{k+1}^{(i+1)}, \Delta u_k^{(i+1)}\}_{k=0}^{N-1}$  at iteration  $i+1$  is

$$\psi = \sum_{k=0}^{N-1} l_k(\Delta x_k, \Delta u_k) + l_N(\Delta x_N) \quad (2.69)$$

in which the stage costs are

$$l_k(\Delta x_k, \Delta u_k) = \frac{1}{2} \Delta x_k' Q_k \Delta x_k + \Delta x_k' M_k \Delta u_k + \frac{1}{2} \Delta u_k' R_k \Delta u_k \quad (2.70a)$$

$$+ q_k' \Delta x_k + r_k' \Delta u_k + f_k, \quad k \in \mathcal{N}$$

$$l_N(\Delta x_N) = \frac{1}{2} \Delta x_N' P_N \Delta x_N + p_N' \Delta x_N + \gamma_N \quad (2.70b)$$

and  $\Delta x_0 = 0$ . The quadratic weight matrices are given by

$$R_0 = W_0^{(i+1)} \quad Q_0 = 0 \quad M_0 = 0 \quad (2.71a)$$

$$\begin{bmatrix} Q_k & M_k \\ M'_k & R_k \end{bmatrix} = W_k^{(i+1)} \quad k = 1, 2, \dots, N-1 \quad (2.71b)$$

$$P_N = W_N^{(i+1)} \quad (2.71c)$$

in which the matrices  $W_k^{(i+1)}$  are computed by (2.68). The parameters related to first and zero order terms of (2.70a) at time index  $k = 0$  are computed by

$$q_0 = 0 \quad (2.72a)$$

$$r_0 = \nabla_{u_0} f_0(x_0, u_0^{(i+1)}) \quad (2.72b)$$

$$f_0 = f_0(x_0, u_0^{(i+1)}) \quad (2.72c)$$

Similarly, the parameters associated with the first and zero order terms of (2.70a) at time indices  $k = 1, 2, \dots, N-1$  are

$$q_k = \nabla_{x_k} f_k(x_k^{(i+1)}, u_k^{(i+1)}) \quad (2.73a)$$

$$r_k = \nabla_{u_k} f_k(x_k^{(i+1)}, u_k^{(i+1)}) \quad (2.73b)$$

$$f_k = f_k(x_k^{(i+1)}, u_k^{(i+1)}) \quad (2.73c)$$

The first and zero order parameters for the stage cost (2.70b) are

$$p_N = \nabla_{x_N} f_N(x_N^{(i+1)}) \quad (2.74a)$$

$$\gamma_N = f_N(x_N^{(i+1)}) \quad (2.74b)$$

*Proof.* The result follows by simple algebraic manipulations of (2.66).  $\square$

**Remark 2.4.6**

The matrices  $R_0$ ,  $\begin{bmatrix} Q_k & M_k \\ M'_k & R_k \end{bmatrix}$  for  $k = 1, 2, \dots, N-1$ , and  $P_N$  computed according to lemma 2.4.5 are all positive definite.

Combination of lemma 2.4.5 and lemma 2.4.1 may be used to formulate the quadratic programs for the SQP algorithm based on structured BFGS approximations of the Hessian matrix. This quadratic program is a constrained linear-quadratic optimal control problem as stated in the following proposition.

**Proposition 2.4.7**

Apply an SQP algorithm using a structured damped BFGS approximation (2.68) of the Hessian to the nonlinear optimal control problem (2.43).

Then the quadratic program for computation of the search direction in the SQP

algorithm applied to (2.43) is

$$\min_{\{\Delta x_{k+1}, \Delta u_k\}} \psi = \sum_{k=0}^{N-1} l_k(\Delta x_k, \Delta u_k) + l_N(\Delta x_N) \quad (2.75a)$$

$$s.t. \quad \Delta x_{k+1} = A'_k \Delta x_k + B'_k \Delta u_k + b_k \quad (2.75b)$$

$$C'_k \Delta x_k + D'_k \Delta u_k + c_k \geq d_k \quad (2.75c)$$

$$C'_N \Delta x_N + c_N \geq d_N \quad (2.75d)$$

in which the initial state is  $\Delta x_0 = 0$ . The objective function is computed according to lemma 2.4.5 and the constraints are computed according to lemma 2.4.1.

*Proof.* Lemma 2.4.1 provides a linearization of the constraints in the nonlinear optimal control problem. Lemma 2.4.5 provides a quadratic approximation of the Lagrangian associated with the nonlinear control problem. The Hessian matrix in this quadratic approximation is a structured BFGS approximation of the Hessian of the Lagrangian of the nonlinear optimal control problem. Consequently, the quadratic program providing the search direction at each iteration of the SQP problem is the stated constrained linear-quadratic optimal control problem. The initial state in the constrained linear-quadratic optimal control problem is  $\Delta x_0 = 0$  as  $x_0$  is a parameter and not a decision variable in the nonlinear optimal control problem.  $\square$

#### Remark 2.4.8

The constrained linear-quadratic optimal control problem (2.75) is strictly convex as the matrices  $R_0$ ,  $\begin{bmatrix} Q_k & M_k \\ M'_k & R_k \end{bmatrix}$  for  $k = 1, 2, \dots, N-1$ , and  $P_N$  all are positive definite by construction.

### 2.4.2 Least Squares Objective

In a broad and important class of nonlinear optimal control problems, the stage costs are weighted least squares functions

$$f_k(x_k, u_k) = \frac{1}{2} \|\varphi_k(x_k, u_k)\|_{\Lambda_k}^2 \quad (2.76a)$$

$$f_N(x_N) = \frac{1}{2} \|\varphi_N(x_N)\|_{\Lambda_N}^2 \quad (2.76b)$$

in which  $\Lambda_k$  for  $k = 0, 1, \dots, N$  are positive semi-definite matrices.

In the classic control problem, the controls  $\{u_k\}_{k=0}^{N-1}$  are selected such that some output functions, say  $\eta_k(x_k, u_k)$ , are as close as possible to a prescribed output trajectory  $\{z_k\}_{k=0}^N$ . In this case, the functions  $\varphi_k$  becomes

$$\varphi_k(x_k, u_k) = \eta_k(x_k, u_k) - z_k \quad k \in \mathcal{N} \quad (2.77a)$$

$$\varphi_N(x_N) = \eta_N(x_N) - z_N \quad (2.77b)$$

Efficient exploitation of the least squares structure of the stage costs, i.e. (2.76), in the nonlinear optimal control problem (2.43) has been described by Biegler (2000) and Bock *et al.* (2000). As in the unconstrained least squares problem (c.f. Dennis and Schnabel, 1996), the least squares structure is exploited by assuming that the residuals,  $\varphi_k$ , are small compared to their derivatives (see appendix A). Under this assumption, the Hessian matrix of the quadratic program in an SQP algorithm may be computed on basis of the first order derivatives of the residuals,  $\nabla\varphi_k$ , rather than the first order derivatives as well as the second order derivatives,  $\nabla^2\varphi_k$ . This approximation of the Hessian matrix for least squares problem under the assumption of small residuals is called the Gauss-Newton approximation.

Let  $w_k$  be defined by (2.53), then the stage costs (2.76) may be expressed as

$$f_k = \frac{1}{2} \|\varphi_k(w_k)\|_{\Lambda_k}^2 \quad k = 0, 1, \dots, N \quad (2.78)$$

The first order derivatives of the objective function (2.43a) with the stage costs (2.76) becomes using the  $w$ -notation

$$\nabla_{w_k} \phi(w) = \nabla f_k(w_k) = \nabla\varphi_k(w_k) \Lambda_k \varphi_k(w_k) \quad (2.79)$$

Invoking the Gauss-Newton approximation (see appendix A) and thereby assuming small residuals, the approximate Hessian matrix of the objective function (2.43a) with the weighted least squares stage costs (2.76) becomes

$$\begin{aligned} \nabla_{w_k w_k}^2 \phi(w) &= \nabla^2 f_k(w_k) \\ &\approx \nabla\varphi_k(w_k) \Lambda_k \nabla\varphi_k(w_k)' \end{aligned} \quad (2.80)$$

The Hessian matrix  $\nabla^2\phi(w)$  has a block diagonal structure as  $\nabla_{w_k w_l}^2 \phi(w) = 0$  for  $k \neq l$ . Hence, the objective function used in the quadratic program of an SQP algorithm applying the Gauss-Newton approximation may be expressed as

$$\begin{aligned} \psi &= \frac{1}{2} \Delta w' W \Delta w + \nabla\phi(w)' \Delta w \\ &= \sum_{k=0}^N \frac{1}{2} \Delta w_k' W_k \Delta w_k + \nabla f_k(w_k)' \Delta w_k \end{aligned} \quad (2.81)$$

in which the diagonal blocks of the Hessian matrix in the quadratic program are computed as

$$W_k = \nabla\varphi_k(w_k) \Lambda_k \nabla\varphi_k(w_k)' \quad (2.82)$$

By construction  $W_k$  is symmetric. Furthermore,  $W_k$  is positive definite, when  $\nabla\varphi_k$  has full row rank and  $\Lambda_k$  is positive definite.

The Gauss-Newton based construction of a quadratic approximation of the Lagrangian associated with the nonlinear optimal control problem is made explicit in the following lemma and proposition. The procedure may be regarded as constructing a quadratic approximation to the objective function in

the nonlinear optimal control problem. Lemma 2.4.9 states the expression for the quadratic approximation of the objective function in the nonlinear optimal control problem. This approximation is identical to the second order approximation of the Lagrangian used in the quadratic program of the SQP algorithm under the Gauss-Newton assumption. Proposition 2.4.10 states the quadratic program obtained by the Gauss-Newton approximation of the nonlinear optimal control problem in an SQP algorithm. This problem is a linear-quadratic optimal control problem.

**Lemma 2.4.9**

Let  $x_0$  be given. Let  $\{x_{k+1}^0, u_k^0\}_{k=0}^{N-1}$  be a trajectory at an iteration of an SQP algorithm applied to (2.43). Let the Hessian of the quadratic program in the SQP algorithm be computed using the Gauss-Newton approximation.

Then the objective function of the quadratic program in an SQP algorithm applied to (2.43) is

$$\psi = \sum_{k=0}^{N-1} l_k(\Delta x_k, \Delta u_k) + l_N(\Delta x_N) \quad (2.83)$$

in which the stage costs are

$$\begin{aligned} l_k(\Delta x_k, \Delta u_k) = & \frac{1}{2} \Delta x_k' Q_k \Delta x_k + \Delta x_k' M_k \Delta u_k \\ & + \frac{1}{2} \Delta u_k' R_k \Delta u_k \\ & + q_k' \Delta x_k + r_k' \Delta u_k + f_k, \quad k \in \mathcal{N} \end{aligned} \quad (2.84a)$$

$$l_N(\Delta x_N) = \frac{1}{2} \Delta x_N' P_N \Delta x_N + p_N' \Delta x_N + \gamma_N \quad (2.84b)$$

and the initial state is  $\Delta x_0 = 0$ . The parameters corresponding to time index  $k = 0$  are

$$Q_0 = \nabla_{x_0} \varphi_0(x_0, u_0^0) \Lambda_0 \nabla_{x_0} \varphi_0(x_0, u_0^0)' \quad (2.85a)$$

$$M_0 = \nabla_{x_0} \varphi_0(x_0, u_0^0) \Lambda_0 \nabla_{u_0} \varphi_0(x_0, u_0^0)' \quad (2.85b)$$

$$R_0 = \nabla_{u_0} \varphi_0(x_0, u_0^0) \Lambda_0 \nabla_{u_0} \varphi_0(x_0, u_0^0)' \quad (2.85c)$$

$$q_0 = \nabla_{x_0} \varphi_0(x_0, u_0^0) \Lambda_0 \varphi_0(x_0, u_0^0) \quad (2.85d)$$

$$r_0 = \nabla_{u_0} \varphi_0(x_0, u_0^0) \Lambda_0 \varphi_0(x_0, u_0^0) \quad (2.85e)$$

$$f_0 = \frac{1}{2} \varphi_0(x_0, u_0^0)' \Lambda_0 \varphi_0(x_0, u_0^0) \quad (2.85f)$$

Similarly, the parameters at time indices  $k = 1, 2, \dots, N-1$  are given by the expressions

$$Q_k = \nabla_{x_k} \varphi_k(x_k^0, u_k^0) \Lambda_k \nabla_{x_k} \varphi_k(x_k^0, u_k^0)' \quad (2.86a)$$

$$M_k = \nabla_{x_k} \varphi_k(x_k^0, u_k^0) \Lambda_k \nabla_{u_k} \varphi_k(x_k^0, u_k^0)' \quad (2.86b)$$

$$R_k = \nabla_{u_k} \varphi_k(x_k^0, u_k^0) \Lambda_k \nabla_{u_k} \varphi_k(x_k^0, u_k^0)' \quad (2.86c)$$

$$q_k = \nabla_{x_k} \varphi_k(x_k^0, u_k^0) \Lambda_k \varphi_k(x_k^0, u_k^0) \quad (2.86d)$$

$$r_k = \nabla_{u_k} \varphi_k(x_k^0, u_k^0) \Lambda_k \varphi_k(x_k^0, u_k^0) \quad (2.86e)$$

$$f_k = \frac{1}{2} \varphi_k(x_k^0, u_k^0)' \Lambda_k \varphi_k(x_k^0, u_k^0) \quad (2.86f)$$

Finally, the parameters at time index  $k = N$  are

$$P_N = \nabla_{x_N} \varphi_N(x_N^0) \Lambda_N \nabla_{x_N} \varphi_N(x_N^0)' \quad (2.87a)$$

$$p_N = \nabla_{x_N} \varphi_N(x_N^0) \Lambda_N \varphi_N(x_N^0) \quad (2.87b)$$

$$\gamma_N = \frac{1}{2} \varphi_N(x_N^0)' \Lambda_N \varphi_N(x_N^0) \quad (2.87c)$$

*Proof.* A quadratic approximation of the objective function of the nonlinear optimal problem is to be developed using the Gauss-Newton approximation.

Consequently, by the discussion in this section the quadratic approximation of the objective function may be expressed as a separable function

$$\psi = \sum_{k=0}^{N-1} l_k(\Delta x_k, \Delta u_k) + l_N(\Delta x_N) \quad (2.88)$$

in which

$$l_0(\Delta x_0, \Delta u_0) = \frac{1}{2} \begin{bmatrix} \Delta x_0 \\ \Delta u_0 \end{bmatrix}' \begin{bmatrix} Q_0 & M_0 \\ M_0' & R_0 \end{bmatrix} \begin{bmatrix} \Delta x_0 \\ \Delta u_0 \end{bmatrix} + \begin{bmatrix} q_0 \\ r_0 \end{bmatrix}' \begin{bmatrix} \Delta x_0 \\ \Delta u_0 \end{bmatrix} + f_0 \quad (2.89a)$$

$$l_k(\Delta x_k, \Delta u_k) = \frac{1}{2} \begin{bmatrix} \Delta x_k \\ \Delta u_k \end{bmatrix}' \begin{bmatrix} Q_k & M_k \\ M_k' & R_k \end{bmatrix} \begin{bmatrix} \Delta x_k \\ \Delta u_k \end{bmatrix} + \begin{bmatrix} q_k \\ r_k \end{bmatrix}' \begin{bmatrix} \Delta x_k \\ \Delta u_k \end{bmatrix} + f_k \quad (2.89b)$$

$$l_N(\Delta x_N) = \frac{1}{2} \Delta x_N' P_N \Delta x_N + p_N' \Delta x_N + \gamma_N \quad (2.89c)$$

for  $k = 1, \dots, N-1$ .

$\Delta x_0$  as  $x_0$  is a parameter and not a decision variable in the nonlinear optimal control problem. The parameters in the quadratic approximation at stage  $k = 0$  of the objective function are

$$f_0 = f_0(x_0, u_0^0) = \frac{1}{2} \|\varphi_0(x_0, u_0^0)\|_{\Lambda_0}^2 = \frac{1}{2} \varphi_0(x_0, u_0^0)' \Lambda_0 \varphi_0(x_0, u_0^0) \quad (2.90a)$$

and

$$q_0 = \nabla_{x_0} \phi = \nabla_{x_0} f_0(x_0, u_0^0) = \nabla_{x_0} \varphi_0(x_0, u_0^0) \Lambda_0 \varphi_0(x_0, u_0^0) \quad (2.90b)$$

$$r_0 = \nabla_{u_0} \phi = \nabla_{u_0} f_0(x_0, u_0^0) = \nabla_{u_0} \varphi_0(x_0, u_0^0) \Lambda_0 \varphi_0(x_0, u_0^0) \quad (2.90c)$$

The parameters associated with the quadratic term is obtained using the Gauss-Newton assumption (see appendix A)

$$\begin{aligned} \begin{bmatrix} Q_0 & M_0 \\ M_0' & R_0 \end{bmatrix} &= W_0 = \nabla \varphi_0(w_0^0) \Lambda_0 \nabla \varphi_0(w_0^0)' \\ &= \begin{bmatrix} \nabla_{x_0} \varphi_0(x_0, u_0^0) \\ \nabla_{u_0} \varphi_0(x_0, u_0^0) \end{bmatrix} \Lambda_0 \begin{bmatrix} \nabla_{x_0} \varphi_0(x_0, u_0^0) \\ \nabla_{u_0} \varphi_0(x_0, u_0^0) \end{bmatrix}' \\ &= \begin{bmatrix} \nabla_{x_0} \varphi_0(x_0, u_0^0) \Lambda_0 \nabla_{x_0} \varphi_0(x_0, u_0^0)' & \nabla_{x_0} \varphi_0(x_0, u_0^0) \Lambda_0 \nabla_{u_0} \varphi_0(x_0, u_0^0)' \\ \nabla_{u_0} \varphi_0(x_0, u_0^0) \Lambda_0 \nabla_{x_0} \varphi_0(x_0, u_0^0)' & \nabla_{u_0} \varphi_0(x_0, u_0^0) \Lambda_0 \nabla_{u_0} \varphi_0(x_0, u_0^0)' \end{bmatrix} \end{aligned} \quad (2.90d)$$

The parameters associated with quadratic approximation of the stage costs at time index  $k = 1, 2, \dots, N-1$  are

$$f_k = f_k(x_k^0, u_k^0) = \frac{1}{2} \|\varphi_k(x_k^0, u_k^0)\|_{\Lambda_k}^2 = \frac{1}{2} \varphi_k(x_k^0, u_k^0)' \Lambda_k \varphi_k(x_k^0, u_k^0) \quad (2.91a)$$

and

$$q_k = \nabla_{x_k} \phi = \nabla_{x_k} f_k(x_k^0, u_k^0) = \nabla_{x_k} \varphi_k(x_k^0, u_k^0) \Lambda_k \varphi_k(x_k^0, u_k^0) \quad (2.91b)$$

$$r_k = \nabla_{u_k} \phi = \nabla_{u_k} f_k(x_k^0, u_k^0) = \nabla_{u_k} \varphi_k(x_k^0, u_k^0) \Lambda_k \varphi_k(x_k^0, u_k^0) \quad (2.91c)$$

as well as the Hessian approximation obtained invoking the Gauss-Newton assumption

$$\begin{aligned} \begin{bmatrix} Q_k & M_k \\ M_k' & R_k \end{bmatrix} &= W_k = \nabla \varphi_k(w_k^0) \Lambda_k \nabla \varphi_k(w_k^0)' \\ &= \begin{bmatrix} \nabla_{x_k} \varphi_k(x_k^0, u_k^0) \\ \nabla_{u_k} \varphi_k(x_k^0, u_k^0) \end{bmatrix} \Lambda_k \begin{bmatrix} \nabla_{x_k} \varphi_k(x_k^0, u_k^0) \\ \nabla_{u_k} \varphi_k(x_k^0, u_k^0) \end{bmatrix}' \\ &= \begin{bmatrix} \nabla_{x_k} \varphi_k(x_k^0, u_k^0) \Lambda_k \nabla_{x_k} \varphi_k(x_k^0, u_k^0)' & \nabla_{x_k} \varphi_k(x_k^0, u_k^0) \Lambda_k \nabla_{u_k} \varphi_k(x_k^0, u_k^0)' \\ \nabla_{u_k} \varphi_k(x_k^0, u_k^0) \Lambda_k \nabla_{x_k} \varphi_k(x_k^0, u_k^0)' & \nabla_{u_k} \varphi_k(x_k^0, u_k^0) \Lambda_k \nabla_{u_k} \varphi_k(x_k^0, u_k^0)' \end{bmatrix} \end{aligned} \quad (2.91d)$$

The parameters associated with quadratic approximation of the stage costs at time index  $k = N$  are

$$\gamma_N = f_N(x_N^0) = \frac{1}{2} \|\varphi_N(x_N^0)\|_{\Lambda_N}^2 = \frac{1}{2} \varphi_N(x_N^0)' \Lambda_N \varphi_N(x_N^0) \quad (2.92a)$$

and

$$p_N = \nabla_{x_N} \phi = \nabla_{x_N} f_N(x_N^0) = \nabla_{x_N} \varphi_N(x_N^0) \Lambda_N \varphi_N(x_N^0) \quad (2.92b)$$

as well as the Hessian approximation obtained invoking the Gauss-Newton assumption

$$\begin{aligned} P_N &= W_N = \nabla_{w_N} \varphi_N(w_N^0) \Lambda_N \nabla_{w_N} \varphi_N(w_N^0)' \\ &= \nabla_{x_N} \varphi_N(x_N^0) \Lambda_N \nabla_{x_N} \varphi_N(x_N^0)' \end{aligned} \quad (2.92c)$$

□

#### Proposition 2.4.10

Let  $x_0$  be given. Let  $\{x_{k+1}, u_k\}_{k=0}^{N-1}$  be a trajectory at an iteration of an SQP algorithm applied to (2.43). Let the Hessian of the quadratic program in the SQP algorithm be computed using the Gauss-Newton approximation.

The quadratic program for generation of a search direction  $\{\Delta x_{k+1}, \Delta u_k\}_{k=0}^{N-1}$  in the SQP algorithm is

$$\min_{\{\Delta x_{k+1}, \Delta u_k\}} \psi = \sum_{k=0}^{N-1} l_k(\Delta x_k, \Delta u_k) + l_N(\Delta x_N) \quad (2.93a)$$

$$s.t. \quad \Delta x_{k+1} = A_k' \Delta x_k + B_k' \Delta u_k + b_k \quad (2.93b)$$

$$C_k' \Delta x_k + D_k' \Delta u_k + c_k \geq d_k \quad (2.93c)$$

$$C_N' \Delta x_N + c_N \geq d_N \quad (2.93d)$$

with the initial state  $\Delta x_0 = 0$ . The stage costs are computed according to lemma 2.4.9 and the constraints are computed according to lemma 2.4.1.

*Proof.* The result follows straightforwardly from lemma 2.4.9 and 2.4.1.  $\Delta x_0 = 0$  as  $x_0$  is a parameter and not a decision variable in the nonlinear optimal control problem.  $\square$

## 2.5 Nonlinear Moving Horizon Estimation

A generic model predictive controller consists of a state estimator, a target calculator, and a regulator. Given new information, i.e. new observations of some process output, the state estimator provides the feedback for the model predictive controller by estimating the current state as well as any exogenous inputs. The target calculator computes the target steady state given an estimate of current exogenous inputs and thereby provides offset free control. Given the current state and the steady state target, the regulator computes the in some sense optimal trajectory from the current state to the target steady state. The first input,  $u_0^*$ , of this trajectory is implemented on the plant, and the cycle is repeated as new information become available.

Early generations of model predictive control technology employed constant or integrating output disturbance feedback (c.f. Qin and Badgwell, 2000), while later feedback schemes have been based on either the Kalman filter or the extended Kalman filter. Recent approaches for state estimation in model predictive control, employ a moving horizon estimator for estimation of the current state and exogenous inputs (c.f. Tenny and Rawlings, 2002; Binder *et al.*, 2001a; Rao and Rawlings, 2000; Robertson *et al.*, 1996).

Assume in the following, that the model consists of two submodels. The first model is a model of the process itself, while the other model is of the disturbance dynamics. The state vector of this augmented model is the process states as well as the states of the disturbance model. The states of the disturbance model provides the basis for computing the exogenous inputs to the process model. The exogenous inputs are used in the target calculator and regulator to obtain offset free behavior of the controlled system.



### 2.5.1 Basic Nonlinear Moving Horizon Estimator

The basic nonlinear moving horizon estimator solves the constrained optimization problem

$$\min_{\{x_k, w_k, v_k\}} \quad \phi = \tilde{l}_0(x_0) + \sum_{k=0}^N \tilde{l}_k(w_k, v_k) \quad (2.94a)$$

$$s.t. \quad x_{k+1} = g_k(x_k, w_k) \quad (2.94b)$$

$$y_k = \varphi_k(x_k) + v_k \quad (2.94c)$$

$$h_k(x_k, w_k) \geq d_k \quad (2.94d)$$

in which the stage costs are of the weighted least squares type

$$\tilde{l}_0(x_0) = \frac{1}{2}(x_0 - \tilde{x}_0)' \tilde{P}_0^{-1}(x_0 - \tilde{x}_0) \quad (2.95a)$$

$$\tilde{l}_k(w_k, v_k) = \frac{1}{2} \begin{bmatrix} w_k - \tilde{w}_k \\ v_k - \tilde{v}_k \end{bmatrix}' \begin{bmatrix} \tilde{Q}_k & \tilde{M}_k \\ \tilde{M}_k' & \tilde{R}_k \end{bmatrix}^{-1} \begin{bmatrix} w_k - \tilde{w}_k \\ v_k - \tilde{v}_k \end{bmatrix} \quad (2.95b)$$

A stochastic interpretation of this objective function, its parameters and their selection is available (c.f. Kailath *et al.*, 2000; Jazwinski, 1970; Robertson *et al.*, 1996). However, this topic is outside the scope of this paper. For the purpose of this paper, it is sufficient to note that the nonlinear moving horizon estimator (2.94) selects the state  $x_0$ , a sequence of process noise  $\{w_k\}_{k=0}^N$  and a sequence of measurement noise  $\{v_k\}_{k=0}^N$  such that the agreement with the measurements  $\{y_k\}_{k=0}^N$  is as good as possible while still respecting the process dynamics (2.94b), the output relation (2.94c), and the constraints (2.94d). The criteria for goodness of fit is specified by the stage costs (2.95). (2.95a) specifies that the state  $x_0$  should preferentially be selected as close to  $\tilde{x}_0$  as possible, while (2.95b) specifies that the noise sequences  $\{w_k, v_k\}_{k=0}^N$  should be as close to  $\{\tilde{w}_k, \tilde{v}_k\}_{k=0}^N$  as possible.

Let  $w_{-1}$  be defined by  $w_{-1} = x_0 - \tilde{x}_0$  and let  $x_{-1} = \tilde{x}_0$ . Then,  $x_0$  may formally be obtained by

$$x_0 = x_{-1} + w_{-1} = g_{-1}(x_{-1}, w_{-1}) \quad (2.96)$$

in which  $g_{-1}$  is defined by this relation and  $x_{-1} = \tilde{x}_0$  is a known parameter. Furthermore, define  $l_{-1}(x_{-1}, w_{-1}) = \tilde{l}_0(x_0)$  such that

$$l_{-1}(x_{-1}, w_{-1}) = \frac{1}{2} w_{-1}' \tilde{P}_0^{-1} w_{-1} \quad (2.97)$$

Use the output relation (2.94c) to express  $v_k$  as a function of  $x_k$

$$v_k = y_k - \varphi_k(x_k) \quad (2.98)$$

such that the stage costs (2.95b) may be expressed as

$$\tilde{l}_k(w_k, v_k) = \tilde{l}_k(w_k, y_k - \varphi_k(x_k)) = l_k(x_k, w_k) \quad (2.99)$$

By these simple rearrangements, it is apparent that the basic nonlinear moving horizon estimator (2.94) may be expressed as

$$\min_{\{x_{k+1}, w_k\}} \phi = \sum_{k=-1}^N l_k(x_k, w_k) \quad (2.100a)$$

$$s.t. \quad x_{k+1} = g_k(x_k, w_k) \quad k = -1, 0, \dots, N \quad (2.100b)$$

$$h_k(x_k, w_k) \geq d_k \quad k = 0, 1, \dots, N \quad (2.100c)$$

This may be regarded as a nonlinear optimal control problem (2.43) in which the initial time index has been shifted from  $k = 0$  to  $k = -1$ , the end time is  $k = N + 1$  rather than  $k = N$ , and the final cost is zero, i.e.  $l_{N+1}(x_{N+1}) = 0$ . Furthermore,  $g_{-1}$  is given by the expression (2.96) rather than by the process dynamics (2.94b) extended to  $k = -1$ .

The next two lemmas, i.e. lemma 2.5.1 and 2.5.2, will in conjunction with the preceding discussion be used to establish proposition 2.5.3. Proposition 2.5.3 expresses the basic nonlinear moving horizon estimator (2.94) as a constrained nonlinear optimal control problem.

**Lemma 2.5.1**

Let  $\begin{bmatrix} \tilde{Q} & \tilde{M} \\ \tilde{M}' & \tilde{R} \end{bmatrix}$  be symmetric and positive definite.

Then the matrices  $W$ ,  $S$ , and  $R$  defined by the relation

$$\begin{bmatrix} W & S' \\ S & V \end{bmatrix} = \begin{bmatrix} \tilde{Q} & \tilde{M} \\ \tilde{M}' & \tilde{R} \end{bmatrix}^{-1} \quad (2.101)$$

exist and are unique. Furthermore, they may be computed by the expressions

$$W = (\tilde{Q} - \tilde{M}\tilde{R}^{-1}\tilde{M}')^{-1} \quad (2.102a)$$

$$S = -\tilde{R}^{-1}\tilde{M}'(\tilde{Q} - \tilde{M}\tilde{R}^{-1}\tilde{M}')^{-1} \quad (2.102b)$$

$$V = (\tilde{R} - \tilde{M}'\tilde{Q}^{-1}\tilde{M})^{-1} \quad (2.102c)$$

*Proof.* Existence and uniqueness as well as symmetry of the inverse (2.101) follows from the assumption of symmetry and positive definiteness of  $\begin{bmatrix} \tilde{Q} & \tilde{M} \\ \tilde{M}' & \tilde{R} \end{bmatrix}$ . (2.102) follows by application of the Schur complement formulas to (2.101) (c.f. Kailath *et al.*, 2000).  $\square$

**Lemma 2.5.2**

Let  $\begin{bmatrix} \tilde{Q}_k & \tilde{M}_k \\ \tilde{M}_k' & \tilde{R}_k \end{bmatrix}$  be symmetric and positive definite for  $k = 0, 1, \dots, N$ . Let  $\tilde{P}_0$  be symmetric and positive definite. Let  $w_{-1} = x_0 - x_{-1}$  and  $x_{-1} = \tilde{x}_0$ . Let  $y_k$  satisfy the output relation (2.94c) of the basic nonlinear moving horizon estimator (2.94).

Then the objective function (2.94a) of the basic nonlinear moving horizon estimator (2.94) may be expressed as

$$\phi = \sum_{k=-1}^N f_k(x_k, w_k) + f_{N+1}(x_{N+1}) \quad (2.103)$$

in which the stage cost is of the weighted least squares type

$$f_k(x_k, w_k) = \frac{1}{2} \|\psi_k(x_k, w_k)\|_{\Lambda_k}^2, \quad k = -1, 0, \dots, N \quad (2.104a)$$

$$f_{N+1}(x_{N+1}) = \frac{1}{2} \|\psi_{N+1}(x_{N+1})\|_{\Lambda_{N+1}}^2 \quad (2.104b)$$

The residual functions  $\psi_k$  are

$$\psi_{-1}(x_{-1}, w_{-1}) = w_{-1} \quad (2.105a)$$

$$\psi_k(x_k, w_k) = \begin{bmatrix} w_k - \tilde{w}_k \\ y_k - \varphi_k(x_k) - \tilde{v}_k \end{bmatrix} \quad k = 0, \dots, N \quad (2.105b)$$

$$\psi_{N+1}(x_{N+1}) = 0 \quad (2.105c)$$

with the weight matrices  $\Lambda_k$  given by

$$\Lambda_{-1} = \tilde{P}_0^{-1} \quad (2.106a)$$

$$\Lambda_k = \begin{bmatrix} W_k & S'_k \\ S_k & V_k \end{bmatrix} = \begin{bmatrix} \tilde{Q}_k & \tilde{M}_k \\ \tilde{M}'_k & \tilde{R}_k \end{bmatrix}^{-1} \quad k = 0, \dots, N \quad (2.106b)$$

$$\Lambda_{N+1} = 0 \quad (2.106c)$$

Furthermore, the block matrices  $W_k$ ,  $S_k$ , and  $V_k$  are given by the relations

$$W_k = \left( \tilde{Q}_k - \tilde{M}_k \tilde{R}_k^{-1} \tilde{M}'_k \right)^{-1} \quad (2.107a)$$

$$S_k = -\tilde{R}_k^{-1} \tilde{M}'_k \left( \tilde{Q}_k - \tilde{M}_k \tilde{R}_k^{-1} \tilde{M}'_k \right)^{-1} \quad (2.107b)$$

$$V_k = \left( \tilde{R}_k - \tilde{M}'_k \tilde{Q}_k^{-1} \tilde{M}_k \right)^{-1} \quad (2.107c)$$

for  $k = 0, 1, \dots, N$ .

*Proof.* The mathematical program (2.100) is equivalent with the mathematical program (2.94) defining the basic nonlinear moving horizon estimator. From inspection of the mathematical program (2.100), it is obvious that the objective function of the basic nonlinear moving horizon estimator (2.94) has the structure

$$\phi = \sum_{k=-1}^N f_k(x_k, w_k) + f_{N+1}(x_{N+1}) \quad (2.108)$$

in which  $f_{N+1}(x_{N+1}) = 0$ .

The stage cost function at time index  $k = -1$  is

$$\begin{aligned} f_{-1}(x_{-1}, w_{-1}) &= l_{-1}(x_{-1}, w_{-1}) = \frac{1}{2} w_{-1} \tilde{P}_0^{-1} w_{-1} \\ &= \frac{1}{2} w_{-1} \Lambda_{-1} w_{-1} = \frac{1}{2} \|w_{-1}\|_{\Lambda_{-1}}^2 = \frac{1}{2} \|\psi_{-1}(x_{-1}, w_{-1})\|_{\Lambda_{-1}}^2 \end{aligned} \quad (2.109)$$

in which the residual function,  $\psi_{-1}$ , is

$$\psi_{-1}(x_{-1}, w_{-1}) = w_{-1} \quad (2.110)$$

and the weight function

$$\Lambda_{-1} = \tilde{P}_0^{-1} \quad (2.111)$$

For  $k = 0, 1, \dots, N$  the stage cost function is

$$\begin{aligned} f_k(x_k, w_k) &= l_k(x_k, w_k) \\ &= \frac{1}{2} \begin{bmatrix} w_k - \tilde{w}_k \\ y_k - \varphi_k(x_k) - \tilde{v}_k \end{bmatrix}' \begin{bmatrix} \tilde{Q}_k & \tilde{M}_k \\ \tilde{M}_k' & \tilde{R}_k \end{bmatrix}^{-1} \begin{bmatrix} w_k - \tilde{w}_k \\ y_k - \varphi_k(x_k) - \tilde{v}_k \end{bmatrix} \\ &= \frac{1}{2} \begin{bmatrix} w_k - \tilde{w}_k \\ y_k - \varphi_k(x_k) - \tilde{v}_k \end{bmatrix}' \begin{bmatrix} W_k & S_k' \\ S_k & V_k \end{bmatrix} \begin{bmatrix} w_k - \tilde{w}_k \\ y_k - \varphi_k(x_k) - \tilde{v}_k \end{bmatrix} \\ &= \frac{1}{2} \|\psi_k(x_k, w_k)\|_{\Lambda_k}^2 \end{aligned} \quad (2.112)$$

in which the expression for the residual function,  $\psi_k$ , is

$$\psi_k(x_k, w_k) = \begin{bmatrix} w_k - \tilde{w}_k \\ y_k - \varphi_k(x_k) - \tilde{v}_k \end{bmatrix} \quad (2.113)$$

The weight matrix,  $\Lambda_k$ , is

$$\Lambda_k = \begin{bmatrix} W_k & S_k' \\ S_k & V_k \end{bmatrix} = \begin{bmatrix} \tilde{Q}_k & \tilde{M}_k \\ \tilde{M}_k' & \tilde{R}_k \end{bmatrix}^{-1} \quad (2.114)$$

in which the block matrices,  $W_k$ ,  $S_k$ , and  $V_k$  may be expressed as

$$W_k = \left( \tilde{Q}_k - \tilde{M}_k \tilde{R}_k^{-1} \tilde{M}_k' \right)^{-1} \quad (2.115a)$$

$$S_k = -\tilde{R}_k^{-1} \tilde{M}_k' \left( \tilde{Q}_k - \tilde{M}_k \tilde{R}_k^{-1} \tilde{M}_k' \right)^{-1} \quad (2.115b)$$

$$V_k = \left( \tilde{R}_k - \tilde{M}_k' \tilde{Q}_k^{-1} \tilde{M}_k \right)^{-1} \quad (2.115c)$$

according to lemma 2.5.1. The stage cost at time index  $k = N + 1$  is

$$f_{N+1}(x_{N+1}) = 0 = \frac{1}{2} \|\psi_{N+1}(x_{N+1})\|_{\Lambda_{N+1}}^2 \quad (2.116)$$

in which  $\psi_{N+1}(x_{N+1}) = 0$  and  $\Lambda_{N+1} = 0$  (or if convenient  $\Lambda_{N+1} = I$ ).  $\square$

**Proposition 2.5.3**

Let  $\begin{bmatrix} \tilde{Q}_k & \tilde{M}_k \\ \tilde{M}_k' & \tilde{R}_k \end{bmatrix}$  be symmetric and positive definite. Let  $\tilde{P}_0$  be symmetric and positive definite.

Then the basic nonlinear moving horizon estimator (2.94) may be expressed as

$$\min_{\{x_{k+1}, w_k\}} \phi = \sum_{k=-1}^N f_k(x_k, w_k) + f_{N+1}(x_{N+1}) \quad (2.117a)$$

$$s.t. \quad x_{k+1} = g_k(x_k, w_k) \quad k = -1, 0, \dots, N \quad (2.117b)$$

$$h_k(x_k, w_k) \geq d_k \quad k = 0, 1, \dots, N \quad (2.117c)$$

The initial state,  $x_{-1}$ , is  $x_{-1} = \tilde{x}_0$  and the function  $g_{-1}(x_{-1}, w_{-1})$  is defined as

$$g_{-1}(x_{-1}, w_{-1}) = x_{-1} + w_{-1} \quad (2.118)$$

The objective function and its stage costs are of the weighted least squares type and given by the relations in lemma 2.5.2.

*Proof.* The result from the equivalence of (2.100) with the basic nonlinear moving horizon estimator (2.94) and lemma 2.5.2.  $\square$

#### Remark 2.5.4

The mathematical program (2.117) has the same structure as the nonlinear optimal control problem (2.43) with a weighted least squares objective function (2.76). Minor differences are present between (2.117) and (2.43). The process noise  $w_k$  is the input in (2.117) while it is the actuators  $u_k$  in (2.43). The initial time is  $k = -1$  in (2.117) while the initial time is  $k = 0$  in (2.43). The final time is  $k = N + 1$  in (2.117) while the final time in (2.43) is  $k = N$ . In the nonlinear optimal control problem (2.43), the inequality constraint,  $h_k(x_k, u_k) \geq d_k$ , is present at the initial time  $k = 0$ , and the inequality constraint  $h_N(x_N) \geq d_N$  associated with the final time  $k = N$  is also present. (2.117) has neither inequality constraint associated with its initial time  $k = -1$  nor any inequality constraint associated with its final time  $k = N + 1$ . However, these minor differences has no implications for the interpretation of (2.117) as a nonlinear optimal control problem (2.43). The initial and final time issue is just a matter of shifting the index. Further, if desirable for the notation, the problem (2.117) can be given inequality constraints at the initial and final time by defining dummy functions  $h_{-1}(x_{-1}, w_{-1}) = 0 \geq -1$  and  $h_{N+1}(x_{N+1}) = 0 \geq -1$ .

By proposition 2.5.3, it is established that the basic nonlinear moving horizon estimator (2.94) is equivalent to the mathematical program (2.117). And (2.117) is an instance of the nonlinear optimal control problem (2.43). Consequently, the basic nonlinear moving horizon estimator (2.94) is an instance of the nonlinear optimal control problem. Furthermore, the objective function in the basic nonlinear moving horizon estimator instance of the optimal control problem is of the weighted least squares type as has been established by lemma 2.5.2. By these arguments it is established, that the basic nonlinear moving horizon estimator (2.94) is an instance of a nonlinear optimal control problem (2.43) with weighted least squares stage costs (2.104) in the objective function (2.117a).

In solving the basic nonlinear moving horizon estimation problem (2.94) by an SQP algorithm, the search direction in each iteration is computed by solving a

linear quadratic optimal control problem (2.30). The linear quadratic optimal control problem providing the search direction is derived from (2.117). The Hessian matrix may be approximated using any of the methods described in section 2.4, as the stage costs are of the weighted least squares type.

In summary, it has been demonstrated that computation of a solution for the basic nonlinear moving horizon estimator (2.94) involves solving linear quadratic optimal control problems (2.30). The objective function of the basic nonlinear moving horizon estimator is of the least squares type. Therefore, the Hessian matrix in the linear quadratic optimal control problem may be computed by either the BFGS method, the method using the Gauss-Newton approximation, or by a hybrid of these two alternatives.

### 2.5.2 Offset Free Nonlinear Moving Horizon Estimator

To have offset free nonlinear model predictive control, Tenny and Rawlings (2002) augmented the model with integrating states which are disturbance terms in either the equation for the process dynamics or the output relation. Muske and Badgwell (2002) suggest the same method to achieve offset free control in the context of linear model predictive control and provide a condition under which this model predictive controller gives offset free control. One of their sufficient conditions for achieving offset free control is that the introduced number of integrating states equals the number of outputs. In a stochastic setting this introduction of integrating states corresponds to letting some of the disturbances be Wiener processes rather than just white noise as in the basic nonlinear moving horizon estimator.

The nonlinear moving horizon estimator augmented with integrating states  $s_k$  to eliminate offset in the model predictive controller is

$$\min_{\{\bar{x}_k, s_k, \bar{w}_k, n_k, \bar{v}_k\}} \phi = \bar{l}_{-1}(\bar{x}_0, s_0) + \sum_{k=0}^N \bar{l}_k(\bar{w}_k, \bar{v}_k, n_k) \quad (2.119a)$$

$$s.t. \quad \bar{x}_{k+1} = \bar{g}_k(\bar{x}_k, \zeta_k, \bar{w}_k) \quad (2.119b)$$

$$s_{k+1} = s_k + n_k \quad (2.119c)$$

$$\zeta_k = G'_k s_k \quad (2.119d)$$

$$\xi_k = H'_k s_k \quad (2.119e)$$

$$y_k = \tilde{\varphi}_k(\bar{x}_k) + \bar{v}_k + \xi_k \quad (2.119f)$$

$$\bar{h}_k(\bar{x}_k, \bar{w}_k) \geq \bar{d}_k \quad (2.119g)$$

$$\eta_k(s_k, n_k) \geq \delta_k \quad (2.119h)$$

in which the stage costs of the objective function are

$$\begin{aligned} \bar{l}_0(\bar{x}_0, s_0) &= \frac{1}{2}(\bar{x}_0 - \hat{x}_0)' \bar{P}_0^{-1}(\bar{x}_0 - \hat{x}_0) \\ &\quad + \frac{1}{2}(s_0 - \tilde{s}_0)' \Pi_0^{-1}(s_0 - \tilde{s}_0) \end{aligned} \quad (2.120a)$$

$$\begin{aligned} \bar{l}_k(\bar{w}_k, n_k, \bar{v}_k) &= \frac{1}{2} \begin{bmatrix} \bar{w}_k - \hat{w}_k \\ \bar{v}_k - \hat{v}_k \end{bmatrix}' \begin{bmatrix} \bar{Q}_k & \bar{M}_k \\ \bar{M}_k' & \bar{R}_k \end{bmatrix}^{-1} \begin{bmatrix} \bar{w}_k - \hat{w}_k \\ \bar{v}_k - \hat{v}_k \end{bmatrix} \\ &\quad + \frac{1}{2}(n_k - \tilde{n}_k)' \Xi_k^{-1}(n_k - \tilde{n}_k) \end{aligned} \quad (2.120b)$$

The barred quantities in (2.119) and (2.120) corresponds to the equivalent quantities in the basic nonlinear moving horizon estimator (2.94).  $s_k$  represents the integrated states which are disturbed by a noise term represented by the vector  $n_k$ . Some linear combination,  $\zeta_k$ , of the integrated states,  $s_k$ , interacts with the original states,  $\bar{x}_k$ , through the relation  $\bar{x}_{k+1} = \bar{g}_k(\bar{x}_k, \zeta_k, \bar{w}_k)$ . These disturbances,  $\zeta_k$ , are called input disturbances. Another, linear combination,  $\xi_k$ , of the integrated states  $s_k$ , interacts with the original variables  $\{\bar{x}_k, \bar{w}_k, \bar{v}_k, y_k\}$  through the output relation  $y_k = \bar{\varphi}_k(\bar{x}_k) + \bar{v}_k + \xi_k$ . These disturbances,  $\xi_k$ , are called output disturbances. Typically, the input and output disturbances are complementary in the sense that the integrated states,  $s_k$ , observable by  $\zeta_k$  are not observable by  $\xi_k$  and vice versa.

**Proposition 2.5.5**

Consider the augmented nonlinear moving horizon estimator (2.119) with the stage costs (2.120). Let  $x_k$ ,  $w_k$ , and  $v_k$  be defined by

$$x_k = \begin{bmatrix} \bar{x}_k \\ s_k \end{bmatrix} \quad w_k = \begin{bmatrix} \bar{w}_k \\ n_k \end{bmatrix} \quad v_k = \bar{v}_k \quad (2.121)$$

Then the augmented nonlinear moving horizon estimator (2.119) with the stage costs (2.120) may be expressed as the mathematical program

$$\min_{\{x_k, w_k, v_k\}} \quad \phi = \tilde{l}_{-1}(x_0) + \sum_{k=0}^N \tilde{l}_k(w_k, v_k) \quad (2.122a)$$

$$s.t. \quad x_{k+1} = g_k(x_k, w_k) \quad k = 0, 1, \dots, N-1 \quad (2.122b)$$

$$y_k = \varphi_k(x_k) + v_k \quad k = 0, 1, \dots, N \quad (2.122c)$$

$$h_k(x_k, w_k) \geq d_k \quad k = 0, 1, \dots, N \quad (2.122d)$$

in which the stage costs are

$$\tilde{l}_{-1}(x_0) = \frac{1}{2}(x_0 - \tilde{x}_0)' \tilde{P}_0^{-1}(x_0 - \tilde{x}_0) \quad (2.123a)$$

$$\tilde{l}_k(w_k, v_k) = \frac{1}{2} \begin{bmatrix} w_k - \tilde{w}_k \\ v_k - \tilde{v}_k \end{bmatrix}' \begin{bmatrix} \tilde{Q}_k & \tilde{M}_k \\ \tilde{M}_k' & \tilde{R}_k \end{bmatrix}^{-1} \begin{bmatrix} w_k - \tilde{w}_k \\ v_k - \tilde{v}_k \end{bmatrix} \quad (2.123b)$$

In the stage cost functions, the weight matrices are

$$\tilde{P}_0 = \begin{bmatrix} \bar{P}_0 & 0 \\ 0 & \Pi_0 \end{bmatrix} \quad (2.124a)$$

$$\tilde{Q}_k = \begin{bmatrix} \bar{Q}_k & 0 \\ 0 & \Xi_k \end{bmatrix} \quad \tilde{M}_k = \begin{bmatrix} \bar{M}_k \\ 0 \end{bmatrix} \quad \tilde{R}_k = \bar{R}_k \quad (2.124b)$$

and the parameter vectors are

$$\tilde{x}_k = \begin{bmatrix} \hat{x}_k \\ \tilde{s}_k \end{bmatrix} \quad \tilde{w}_k = \begin{bmatrix} \hat{w}_k \\ \tilde{n}_k \end{bmatrix} \quad \tilde{v}_k = \hat{v}_k \quad (2.125)$$

The state transition function  $g_k$  is

$$g_k(x_k, w_k) = \begin{bmatrix} \tilde{g}_k(\bar{x}_k, G'_k s_k, \bar{w}_k) \\ s_k + n_k \end{bmatrix} \quad (2.126)$$

and the output map  $\varphi_k$  is

$$\varphi_k(x_k) = \tilde{\varphi}_k(\bar{x}_k) + H'_k s_k \quad (2.127)$$

The function  $h_k$  and the lower value  $d_k$  defining the inequality constraints are

$$h_k(x_k, w_k) = \begin{bmatrix} \bar{h}_k(\bar{x}_k, \bar{w}_k) \\ \eta_k(s_k, n_k) \end{bmatrix} \quad d_k = \begin{bmatrix} \bar{d}_k \\ \delta_k \end{bmatrix} \quad (2.128)$$

*Proof.* The result follows by noting that the stage costs of (2.119) may be expressed as

$$\begin{aligned} \bar{l}_0(\bar{x}_0, s_0) &= \frac{1}{2}(\bar{x}_0 - \hat{x}_0)' \bar{P}_0^{-1}(\bar{x}_0 - \hat{x}_0) + \frac{1}{2}(s_0 - \tilde{s}_0)' \Pi_0^{-1}(s_0 - \tilde{s}_0) \\ &= \frac{1}{2} \begin{bmatrix} \bar{x}_0 - \hat{x}_0 \\ s_0 - \tilde{s}_0 \end{bmatrix}' \begin{bmatrix} \bar{P}_0 & 0 \\ 0 & \Pi_0 \end{bmatrix}^{-1} \begin{bmatrix} \bar{x}_0 - \hat{x}_0 \\ s_0 - \tilde{s}_0 \end{bmatrix} \\ &= \frac{1}{2}(x_0 - \tilde{x}_0)' P_0^{-1}(x_0 - \tilde{x}_0) = \tilde{l}_{-1}(x_0) \end{aligned} \quad (2.129)$$

and for  $k = 0, 1, \dots, N$

$$\begin{aligned} \bar{l}_k(\bar{w}_k, n_k, \bar{v}_k) &= \frac{1}{2} \begin{bmatrix} \bar{w}_k - \hat{w}_k \\ \bar{v}_k - \hat{v}_k \end{bmatrix}' \begin{bmatrix} \tilde{Q}_k & \tilde{M}_k \\ \tilde{M}'_k & \tilde{R}_k \end{bmatrix}^{-1} \begin{bmatrix} \bar{w}_k - \hat{w}_k \\ \bar{v}_k - \hat{v}_k \end{bmatrix} \\ &\quad + \frac{1}{2}(n_k - \tilde{n}_k)' \Xi_k^{-1}(n_k - \tilde{n}_k) \\ &= \frac{1}{2} \begin{bmatrix} \bar{w}_k - \hat{w}_k \\ \bar{v}_k - \hat{v}_k \\ n_k - \tilde{n}_k \end{bmatrix}' \begin{bmatrix} \tilde{Q}_k & \tilde{M}_k & 0 \\ \tilde{M}'_k & \tilde{R}_k & 0 \\ 0 & 0 & \Xi_k \end{bmatrix}^{-1} \begin{bmatrix} \bar{w}_k - \hat{w}_k \\ \bar{v}_k - \hat{v}_k \\ n_k - \tilde{n}_k \end{bmatrix} \\ &= \frac{1}{2} \begin{bmatrix} \bar{w}_k - \hat{w}_k \\ n_k - \tilde{n}_k \\ \bar{v}_k - \hat{v}_k \end{bmatrix}' \begin{bmatrix} \tilde{Q}_k & 0 & \tilde{M}_k \\ 0 & \Xi_k & 0 \\ \tilde{M}'_k & 0 & \tilde{R}_k \end{bmatrix}^{-1} \begin{bmatrix} \bar{w}_k - \hat{w}_k \\ n_k - \tilde{n}_k \\ \bar{v}_k - \hat{v}_k \end{bmatrix} \\ &= \frac{1}{2} \begin{bmatrix} w_k - \tilde{w}_k \\ v_k - \tilde{v}_k \end{bmatrix}' \begin{bmatrix} \tilde{Q}_k & \tilde{M}_k \\ \tilde{M}'_k & \tilde{R}_k \end{bmatrix}^{-1} \begin{bmatrix} w_k - \tilde{w}_k \\ v_k - \tilde{v}_k \end{bmatrix} = \tilde{l}_k(w_k, v_k) \end{aligned} \quad (2.130)$$

in which

$$\begin{aligned} \tilde{P}_0 &= \begin{bmatrix} \bar{P}_0 & 0 \\ 0 & \Pi_0 \end{bmatrix} & \tilde{Q}_k &= \begin{bmatrix} \tilde{Q}_k & 0 \\ 0 & \Xi_k \end{bmatrix} & \tilde{M}_k &= \begin{bmatrix} \tilde{M}_k \\ 0 \end{bmatrix} & \tilde{R}_k &= \tilde{R}_k \\ x_0 &= \begin{bmatrix} \bar{x}_0 \\ s_0 \end{bmatrix} & \tilde{x}_0 &= \begin{bmatrix} \hat{x}_0 \\ \tilde{s}_0 \end{bmatrix} & w_k &= \begin{bmatrix} \bar{w}_k \\ n_k \end{bmatrix} & \tilde{w}_k &= \begin{bmatrix} \hat{w}_k \\ \tilde{n}_k \end{bmatrix} \end{aligned}$$



The constraints

$$\bar{x}_{k+1} = \bar{g}_k(\bar{x}_k, \zeta_k, \bar{w}_k) \quad (2.131a)$$

$$s_{k+1} = s_k + n_k \quad (2.131b)$$

$$\zeta_k = G'_k s_k \quad (2.131c)$$

becomes by elimination of  $\zeta_k$

$$\begin{bmatrix} \bar{x}_{k+1} \\ s_{k+1} \end{bmatrix} = \begin{bmatrix} \bar{g}_k(\bar{x}_k, G'_k s_k, \bar{w}_k) \\ s_k + n_k \end{bmatrix} \quad (2.132)$$

which may be expressed as

$$x_{k+1} = g_k(x_k, w_k) \quad (2.133)$$

in which

$$x_k = \begin{bmatrix} \bar{x}_k \\ s_k \end{bmatrix} \quad g_k(x_k, w_k) = \begin{bmatrix} \bar{g}_k(\bar{x}_k, G'_k s_k, \bar{w}_k) \\ s_k + n_k \end{bmatrix}$$

The output disturbance,  $\xi_k = H'_k s_k$ , may be used to express the measurement,  $y_k$ , as

$$\begin{aligned} y_k &= \tilde{\varphi}_k(\bar{x}_k) + \bar{v}_k + \xi_k \\ &= \tilde{\varphi}_k(\bar{x}_k) + \bar{v}_k + H'_k s_k \\ &= \varphi_k(x_k) + v_k \end{aligned} \quad (2.134)$$

in which

$$\varphi_k(x_k) = \tilde{\varphi}_k(\bar{x}_k) + H'_k s_k \quad (2.135)$$

The constraints

$$\bar{h}_k(\bar{x}_k, \bar{w}_k) \geq \bar{d}_k \quad (2.136a)$$

$$\eta_k(s_k, n_k) \geq \delta_k \quad (2.136b)$$

may be expressed as

$$h_k(x_k, w_k) \geq d_k \quad (2.137)$$

in which

$$h_k(x_k, w_k) = \begin{bmatrix} \bar{h}_k(\bar{x}_k, \bar{w}_k) \\ \eta_k(s_k, n_k) \end{bmatrix} \quad d_k = \begin{bmatrix} \bar{d}_k \\ \delta_k \end{bmatrix}$$

Consequently, the augmented nonlinear moving horizon estimator (2.119) may be expressed as

$$\min_{\{x_k, w_k, v_k\}} \quad \phi = \tilde{l}_{-1}(x_0) + \sum_{k=0}^N \tilde{l}_k(w_k, v_k) \quad (2.138a)$$

$$s.t. \quad x_{k+1} = g_k(x_k, w_k) \quad k = 0, 1, \dots, N-1 \quad (2.138b)$$

$$y_k = \varphi_k(x_k) + v_k \quad k = 0, 1, \dots, N \quad (2.138c)$$

$$h_k(x_k, w_k) \geq d_k \quad k = 0, 1, \dots, N \quad (2.138d)$$

in which the quantities are defined as outlined in this proof and stated in the proposition.  $\square$

By proposition 2.5.5, it is established that the augmented nonlinear moving horizon estimator (2.119) for offset free model predictive control may be expressed as a basic nonlinear moving horizon estimator (2.94). Hence, computation of the estimate prescribed by (2.119) corresponds to solving a nonlinear moving horizon estimation problem which in turn is equivalent to a nonlinear optimal control problem. Solution of the nonlinear optimal control problem by an SQP algorithm requires computation of a search direction by solution of a linear quadratic optimal control problem (2.30). In conclusion, efficient solution of the offset free nonlinear moving horizon estimation problem by an SQP algorithm involves solution of a constrained linear-quadratic optimal control problem.

### 2.5.3 Moving Horizon Estimation

Robertson *et al.* (1996) provide a statistical interpretation of the moving horizon estimator with linear models, and Muske and Badgwell (2002) give sufficient requirements for the moving horizon estimator with linear models to yield offset free model predictive control. The moving horizon estimator enabling offset free model predictive control will be called the augmented moving horizon estimator, as the mechanism for enabling offset free model predictive control is to augment the state vector with integrated states that are either input or output disturbances.

The augmented moving horizon estimator with a linear model is

$$\min_{\substack{\{\bar{x}_k, \bar{w}_k, \bar{v}_k, \\ s_k, n_k, \zeta_k, \xi_k\}}} \phi = \bar{l}_{-1}(\bar{x}_0, s_0) + \sum_{k=0}^N \bar{l}_k(\bar{w}_k, n_k, \bar{v}_k) \quad (2.139a)$$

$$s.t. \quad \bar{x}_{k+1} = \bar{A}'_k \bar{x}_k + \bar{B}'_k \bar{w}_k + \bar{b}_k + \zeta_k \quad k \in \mathcal{N}_E \quad (2.139b)$$

$$s_{k+1} = s_k + n_k \quad k \in \mathcal{N}_E \quad (2.139c)$$

$$\zeta_k = \bar{G}'_k s_k \quad k \in \mathcal{N}_E \quad (2.139d)$$

$$\xi_k = \bar{H}'_k s_k \quad k \in \mathcal{N}_E \quad (2.139e)$$

$$y_k = \bar{C}'_k \bar{x}_k + \bar{c}_k + \bar{v}_k + \xi_k \quad k \in \mathcal{N}_E \quad (2.139f)$$

$$\bar{S}'_k \bar{x}_k + \bar{T}_k \bar{w}_k + t_k \geq d_k \quad k \in \mathcal{N}_E \quad (2.139g)$$

$$\bar{U}'_k s_k + \bar{V}'_k n_k + \rho_k \geq \delta_k \quad k \in \mathcal{N}_E \quad (2.139h)$$

in which  $\mathcal{N}_E = \{0, 1, \dots, N\}$ . In this model, the dynamics (2.139b) is augmented with the dynamics (2.139c) integrating the states,  $s_k$ . A linear combination,  $\zeta_k$ , of these states,  $s_k$ , are input or state disturbances as they affect the original variables through (2.139b). Another, linear combination,  $\xi_k$ , of the integrated states,  $s_k$ , are output disturbances as they affect the original variables through (2.139f). The original states,  $\bar{x}_k$ , and the original process noise,  $\bar{w}_k$ , are constrained by the inequality constraint (2.139g). In a statistical context, these inequality constraints may be used to specify truncated probability distributions (c.f. Rao, 2000). Similar inequality constraints (2.139h) are

introduced for the augmented states,  $s_k$ , and the augmented process noise,  $n_k$ . The stage costs,  $\bar{l}_k$ , of the objective function (2.139a) are defined as

$$\begin{aligned} \bar{l}_{-1}(\bar{x}_0, s_0) &= \frac{1}{2}(\bar{x}_0 - \hat{x}_0)' \bar{P}_0^{-1}(\bar{x}_0 - \hat{x}_0) \\ &\quad + \frac{1}{2}(s_0 - \tilde{s}_0)' \Pi_0^{-1}(s_0 - \tilde{s}_0) \end{aligned} \quad (2.140a)$$

$$\begin{aligned} \bar{l}_k(\bar{w}_k, n_k, \bar{v}_k) &= \frac{1}{2} \begin{bmatrix} \bar{w}_k - \hat{w}_k \\ \bar{v}_k - \hat{v}_k \end{bmatrix}' \begin{bmatrix} \bar{Q}_k & \bar{M}_k \\ \bar{M}_k' & \bar{R}_k \end{bmatrix}^{-1} \begin{bmatrix} \bar{w}_k - \hat{w}_k \\ \bar{v}_k - \hat{v}_k \end{bmatrix} \\ &\quad + \frac{1}{2}(n_k - \tilde{n}_k)' \Xi_k^{-1}(n_k - \tilde{n}_k) \end{aligned} \quad k \in \mathcal{N}_E \quad (2.140b)$$

in which  $\bar{P}_0$ ,  $\Pi_0$ ,  $\begin{bmatrix} \bar{Q}_k & \bar{M}_k \\ \bar{M}_k' & \bar{R}_k \end{bmatrix}$ , and  $\Xi_k$  are assumed to be symmetric and positive definite. The decision variables  $\{\bar{x}_k, s_k, \bar{w}_k, n_k, \bar{v}_k, \zeta_k, \xi_k\}$  should be selected such that they respect the constraints (2.139b)-(2.139h) with the given measurements  $\{y_k\}_{k=0}^N$  and such that  $\bar{x}_0$  is close to  $\hat{x}_0$ ,  $s_0$  is close to  $\tilde{s}_0$ ,  $\bar{w}_k$  is close to  $\hat{w}_k$ ,  $n_k$  is close to  $\tilde{n}_k$ , and  $\bar{v}_k$  is close to  $\hat{v}_k$ . A probabilistic analysis of the moving horizon estimation problem addresses the optimal selection of the parameters in the stage cost functions. However, this topic is outside the scope of this paper.

The next proposition states the augmented moving horizon estimator for linear models (2.139) as a basic moving horizon estimator for linear models (2.142). The estimator is equivalent to the basic nonlinear moving horizon estimator (2.94) except that the involved equations in (2.142) are affine rather than general functions.

**Proposition 2.5.6**

Consider the augmented moving horizon estimator with affine constraints (2.139). Let

$$x_k = \begin{bmatrix} \bar{x}_k \\ s_k \end{bmatrix} \quad w_k = \begin{bmatrix} \bar{w}_k \\ n_k \end{bmatrix} \quad v_k = \bar{v}_k \quad (2.141)$$

Then the augmented moving horizon estimator with affine constraints (2.139) may be expressed as

$$\min_{\{x_k, w_k, v_k\}} \quad \phi = \tilde{l}_{-1}(x_0) + \sum_{k=0}^N \tilde{l}_k(w_k, v_k) \quad (2.142a)$$

$$s.t. \quad x_{k+1} = \tilde{A}_k' x_k + \tilde{B}_k' w_k + \tilde{b}_k \quad k \in \mathcal{N}_E \quad (2.142b)$$

$$y_k = \tilde{C}_k' x_k + \tilde{c}_k + v_k \quad k \in \mathcal{N}_E \quad (2.142c)$$

$$\tilde{E}_k' x_k + \tilde{F}_k' w_k + \tilde{h}_k \geq \tilde{d}_k \quad k \in \mathcal{N}_E \quad (2.142d)$$

in which the stage costs are

$$\tilde{l}_{-1}(x_0) = \frac{1}{2}(x_0 - \tilde{x}_0)' \tilde{P}_0^{-1}(x_0 - \tilde{x}_0) \quad (2.143a)$$

$$\tilde{l}_k(w_k, v_k) = \frac{1}{2} \begin{bmatrix} w_k - \tilde{w}_k \\ v_k - \tilde{v}_k \end{bmatrix}' \begin{bmatrix} \tilde{Q}_k & \tilde{M}_k \\ \tilde{M}_k' & \tilde{R}_k \end{bmatrix}^{-1} \begin{bmatrix} w_k - \tilde{w}_k \\ v_k - \tilde{v}_k \end{bmatrix} \quad k \in \mathcal{N}_E \quad (2.143b)$$

The parameters in the stage cost functions (2.143) are

$$\tilde{P}_0 = \begin{bmatrix} \bar{P}_0 & 0 \\ 0 & \Pi_0 \end{bmatrix} \quad (2.144a)$$

$$\tilde{Q}_k = \begin{bmatrix} \bar{Q}_k & 0 \\ 0 & \Xi_k \end{bmatrix} \quad \tilde{M}_k = \begin{bmatrix} \bar{M}_k \\ 0 \end{bmatrix} \quad \tilde{R}_k = \bar{R}_k \quad k \in \mathcal{N}_E \quad (2.144b)$$

$$\tilde{x}_k = \begin{bmatrix} \hat{x}_k \\ \tilde{s}_k \end{bmatrix} \quad \tilde{w}_k = \begin{bmatrix} \hat{w}_k \\ \tilde{n}_k \end{bmatrix} \quad \tilde{v}_k = \hat{v}_k \quad (2.144c)$$

and the parameters of the constraints (2.142b)-(2.142d) are

$$\tilde{A}'_k = \begin{bmatrix} \bar{A}'_k & \bar{G}'_k \\ 0 & I \end{bmatrix} \quad \tilde{B}'_k = \begin{bmatrix} \bar{B}'_k & 0 \\ 0 & I \end{bmatrix} \quad \tilde{b}_k = \begin{bmatrix} \bar{b}_k \\ 0 \end{bmatrix} \quad k \in \mathcal{N}_E \quad (2.145a)$$

$$\tilde{C}'_k = \begin{bmatrix} \bar{C}'_k & \bar{H}'_k \end{bmatrix} \quad \tilde{c}_k = \bar{c}_k \quad k \in \mathcal{N}_E \quad (2.145b)$$

$$\tilde{E}'_k = \begin{bmatrix} \bar{S}'_k & 0 \\ 0 & \bar{U}'_k \end{bmatrix} \quad \tilde{F}'_k = \begin{bmatrix} \bar{T}'_k & 0 \\ 0 & \bar{V}'_k \end{bmatrix} \quad \tilde{h}_k = \begin{bmatrix} t_k \\ \rho_k \end{bmatrix} \quad \tilde{d}_k = \begin{bmatrix} \bar{d}_k \\ \delta_k \end{bmatrix} \quad k \in \mathcal{N}_E \quad (2.145c)$$

*Proof.* The result follows immediately by simple algebraic manipulations. The stage cost function,  $\tilde{l}_{-1}$ , is obtained as

$$\begin{aligned} \bar{l}_{-1}(\bar{x}_0, s_0) &= \frac{1}{2}(\bar{x}_0 - \hat{x}_0)' \bar{P}_0^{-1}(\bar{x}_0 - \hat{x}_0) + \frac{1}{2}(s_0 - \tilde{s}_0)' \Pi_0^{-1}(s_0 - \tilde{s}_0) \\ &= \frac{1}{2} \begin{bmatrix} \bar{x}_0 - \hat{x}_0 \\ s_0 - \tilde{s}_0 \end{bmatrix}' \begin{bmatrix} \bar{P}_0 & 0 \\ 0 & \Pi_0 \end{bmatrix}^{-1} \begin{bmatrix} \bar{x}_0 - \hat{x}_0 \\ s_0 - \tilde{s}_0 \end{bmatrix} \\ &= \frac{1}{2}(x_0 - \tilde{x}_0)' \tilde{P}_0^{-1}(x_0 - \tilde{x}_0) = \tilde{l}_{-1}(x_0) \end{aligned} \quad (2.146)$$

in which

$$\tilde{P}_0 = \begin{bmatrix} \bar{P}_0 & 0 \\ 0 & \Pi_0 \end{bmatrix} \quad x_0 = \begin{bmatrix} \bar{x}_0 \\ s_0 \end{bmatrix} \quad \tilde{x}_0 = \begin{bmatrix} \hat{x}_0 \\ \tilde{s}_0 \end{bmatrix}$$

Similarly, at time index  $k \in \mathcal{N}_E = \{0, 1, \dots, N\}$  we obtain

$$\begin{aligned} \bar{l}_k(\bar{w}_k, n_k, \bar{v}_k) &= \frac{1}{2} \begin{bmatrix} \bar{w}_k - \hat{w}_k \\ \bar{v}_k - \hat{v}_k \end{bmatrix}' \begin{bmatrix} \bar{Q}_k & \bar{M}_k \\ \bar{M}'_k & \bar{R}_k \end{bmatrix}^{-1} \begin{bmatrix} \bar{w}_k - \hat{w}_k \\ \bar{v}_k - \hat{v}_k \end{bmatrix} \\ &\quad + \frac{1}{2}(n_k - \tilde{n}_k)' \Xi_k^{-1}(n_k - \tilde{n}_k) \\ &= \frac{1}{2} \begin{bmatrix} \bar{w}_k - \hat{w}_k \\ \bar{v}_k - \hat{v}_k \\ n_k - \tilde{n}_k \end{bmatrix}' \begin{bmatrix} \bar{Q}_k & \bar{M}_k & 0 \\ \bar{M}'_k & \bar{R}_k & 0 \\ 0 & 0 & \Xi_k \end{bmatrix}^{-1} \begin{bmatrix} \bar{w}_k - \hat{w}_k \\ \bar{v}_k - \hat{v}_k \\ n_k - \tilde{n}_k \end{bmatrix} \\ &= \frac{1}{2} \begin{bmatrix} \bar{w}_k - \hat{w}_k \\ n_k - \tilde{n}_k \\ \bar{v}_k - \hat{v}_k \end{bmatrix}' \begin{bmatrix} \bar{Q}_k & 0 & \bar{M}_k \\ 0 & \Xi_k & 0 \\ \bar{M}'_k & 0 & \bar{R}_k \end{bmatrix}^{-1} \begin{bmatrix} \bar{w}_k - \hat{w}_k \\ n_k - \tilde{n}_k \\ \bar{v}_k - \hat{v}_k \end{bmatrix} \\ &= \frac{1}{2} \begin{bmatrix} w_k - \tilde{w}_k \\ v_k - \tilde{v}_k \end{bmatrix}' \begin{bmatrix} \tilde{Q}_k & \tilde{M}_k \\ \tilde{M}'_k & \tilde{R}_k \end{bmatrix}^{-1} \begin{bmatrix} w_k - \tilde{w}_k \\ v_k - \tilde{v}_k \end{bmatrix} = \tilde{l}_k(w_k, v_k) \end{aligned} \quad (2.147)$$

in which we have introduced and defined the quantities

$$\begin{aligned} w_k &= \begin{bmatrix} \bar{w}_k \\ n_k \end{bmatrix} & \tilde{w}_k &= \begin{bmatrix} \hat{w}_k \\ \tilde{n}_k \end{bmatrix} & v_k &= \bar{v}_k & \tilde{v}_k &= \hat{v}_k \\ \tilde{Q}_k &= \begin{bmatrix} \bar{Q}_k & 0 \\ 0 & \Xi_k \end{bmatrix} & \tilde{M}_k &= \begin{bmatrix} \bar{M}_k \\ 0 \end{bmatrix} & \tilde{R}_k &= \bar{R}_k \end{aligned}$$

The constraints for  $k \in \mathcal{N}_E$

$$\bar{x}_{k+1} = \bar{A}'_k \bar{x}_k + \bar{B}'_k \bar{w}_k + \bar{b}_k + \zeta_k \quad (2.148a)$$

$$s_{k+1} = s_k + n_k \quad (2.148b)$$

$$\zeta_k = \bar{G}'_k s_k \quad (2.148c)$$

may by substitution of the expressions for  $\zeta_k$  in (2.148a) and subsequent vectorization by stacking be expressed as

$$\begin{bmatrix} \bar{x}_{k+1} \\ s_{k+1} \end{bmatrix} = \begin{bmatrix} \bar{A}'_k & \bar{G}'_k \\ 0 & I \end{bmatrix} \begin{bmatrix} \bar{x}_k \\ s_k \end{bmatrix} + \begin{bmatrix} \bar{B}'_k & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \bar{w}_k \\ n_k \end{bmatrix} + \begin{bmatrix} \bar{b}_k \\ 0 \end{bmatrix} \quad (2.149)$$

Let

$$x_k = \begin{bmatrix} \bar{x}_k \\ s_k \end{bmatrix} \quad (2.150)$$

Then (2.149) may be expressed as

$$x_{k+1} = \tilde{A}'_k x_k + \tilde{B}'_k w_k + \tilde{b}_k \quad k \in \mathcal{N}_E \quad (2.151)$$

in which

$$\tilde{A}'_k = \begin{bmatrix} \bar{A}'_k & \bar{G}'_k \\ 0 & I \end{bmatrix} \quad \tilde{B}'_k = \begin{bmatrix} \bar{B}'_k & 0 \\ 0 & I \end{bmatrix} \quad \tilde{b}_k = \begin{bmatrix} \bar{b}_k \\ 0 \end{bmatrix} \quad k \in \mathcal{N}_E$$

The relation for the measurement,  $y_k$ , may be formulated as

$$\begin{aligned} y_k &= \bar{C}'_k \bar{x}_k + \bar{c}_k + \bar{v}_k + \xi_k \\ &= \bar{C}'_k \bar{x}_k + \bar{c}_k + \bar{v}_k + \bar{H}'_k s_k \\ &= [\bar{C}'_k \quad \bar{H}'_k] \begin{bmatrix} \bar{x}_k \\ s_k \end{bmatrix} + \bar{c}_k + \bar{v}_k \\ &= \tilde{C}'_k x_k + \tilde{c}_k + v_k \end{aligned} \quad k \in \mathcal{N}_E \quad (2.152)$$

in which we have introduced and defined the quantities  $\tilde{C}'_k$  and  $\tilde{c}_k$  by

$$\tilde{C}'_k = [\bar{C}'_k \quad \bar{H}'_k] \quad \tilde{c}_k = \bar{c}_k \quad k \in \mathcal{N}_E$$

The constraints for  $k \in \mathcal{N}_E$

$$\bar{S}'_k \bar{x}_k + \bar{T}'_k \bar{w}_k + t_k \geq d_k \quad (2.153a)$$

$$\bar{U}'_k s_k + \bar{V}'_k n_k + \rho_k \geq \delta_k \quad (2.153b)$$

may be stacked to give the expression

$$\begin{bmatrix} \bar{S}'_k & 0 \\ 0 & \bar{U}'_k \end{bmatrix} \begin{bmatrix} \bar{x}_k \\ s_k \end{bmatrix} + \begin{bmatrix} \bar{T}'_k & 0 \\ 0 & \bar{V}'_k \end{bmatrix} \begin{bmatrix} \bar{w}_k \\ n_k \end{bmatrix} + \begin{bmatrix} t_k \\ \rho_k \end{bmatrix} \geq \begin{bmatrix} d_k \\ \rho_k \end{bmatrix} \quad (2.154)$$

that may be reformulated as

$$\tilde{E}'_k x_k + \tilde{F}'_k w_k + \tilde{h}_k \geq \tilde{d}_k \quad k \in \mathcal{N}_E \quad (2.155)$$

by definition of the quantities

$$\tilde{E}_k = \begin{bmatrix} \bar{S}_k & 0 \\ 0 & \bar{U}_k \end{bmatrix} \quad \tilde{F}_k = \begin{bmatrix} \bar{T}_k & 0 \\ 0 & \bar{V}_k \end{bmatrix} \quad \tilde{h}_k = \begin{bmatrix} t_k \\ \rho_k \end{bmatrix} \quad \tilde{d}_k = \begin{bmatrix} d_k \\ \rho_k \end{bmatrix} \quad k \in \mathcal{N}_E$$

□

(2.142) is called the basic moving horizon estimator for linear models. In the case with no inequality constraints, the problem is equivalent to the smoothing problem. Though smoothing problems are often stated for linear rather than affine models (c.f. Kailath *et al.*, 2000). By proposition 2.5.6 it is established how to convert the augmented moving horizon estimator with linear models (2.139) to a basic moving horizon estimator for linear models (2.142).

The next proposition establishes a procedure for converting the basic moving horizon estimator for linear models (2.142) to a least squares constrained optimal control problem for linear models (2.156).

**Proposition 2.5.7**

Consider the basic linear moving horizon estimator (2.142). Let  $\tilde{P}_0$  and  $\begin{bmatrix} \tilde{Q}_k & \tilde{M}_k \\ \tilde{M}'_k & \tilde{R}_k \end{bmatrix}$  be symmetric and positive definite.

Then the basic linear moving horizon estimator (2.142) is identical with the linear least squares constrained optimal control problem

$$\min_{\{x_{k+1}, w_k\}} \phi = \sum_{k=-1}^N l_k(x_k, w_k) + l_{N+1}(x_{N+1}) \quad (2.156a)$$

$$s.t. \quad x_{k+1} = A'_k x_k + B'_k w_k + b_k \quad k \in \{-1\} \cup \mathcal{N}_E \quad (2.156b)$$

$$C'_k x_k + D'_k w_k + c_k \geq d_k \quad k \in \{-1\} \cup \mathcal{N}_E \quad (2.156c)$$

$$C'_{N+1} x_{N+1} + c_{N+1} \geq d_{N+1} \quad (2.156d)$$

in which the stage costs are of the least squares type

$$l_{-1}(x_{-1}, w_{-1}) = \frac{1}{2} \|\psi_{-1}(x_{-1}, w_{-1})\|_{\Lambda_{-1}}^2 \quad (2.157a)$$

$$l_k(x_k, w_k) = \frac{1}{2} \|\psi_k(x_k, w_k)\|_{\Lambda_k}^2 \quad k \in \mathcal{N}_E \quad (2.157b)$$

$$l_{N+1}(x_{N+1}) = \frac{1}{2} \|\psi_{N+1}(x_{N+1})\|_{\Lambda_{N+1}}^2 \quad (2.157c)$$

and the residual functions are

$$\psi_{-1}(x_{-1}, w_{-1}) = w_{-1} \quad (2.158a)$$

$$\psi_k(x_k, w_k) = \begin{bmatrix} w_k - \tilde{w}_k \\ y_k - \tilde{C}'_k x_k - \tilde{c}_k - \tilde{v}_k \end{bmatrix} \quad k \in \mathcal{N}_E \quad (2.158b)$$

$$\psi_{N+1}(x_{N+1}) = 0 \quad (2.158c)$$

The weight matrices  $\Lambda_k$  are

$$\Lambda_{-1} = \tilde{P}_0^{-1} \quad (2.159a)$$

$$\Lambda_k = \begin{bmatrix} W_k & S'_k \\ S_k & V_k \end{bmatrix} = \begin{bmatrix} \tilde{Q}_k & \tilde{M}_k \\ \tilde{M}'_k & \tilde{R}_k \end{bmatrix}^{-1} \quad k \in \mathcal{N}_E \quad (2.159b)$$

$$\Lambda_{N+1} = 0 \quad (2.159c)$$

in which

$$W_k = \left( \tilde{Q}_k - \tilde{M}_k \tilde{R}_k^{-1} \tilde{M}'_k \right)^{-1} \quad (2.160a)$$

$$S_k = -\tilde{R}_k^{-1} \tilde{M}'_k \left( \tilde{Q}_k - \tilde{M}_k \tilde{R}_k^{-1} \tilde{M}'_k \right)^{-1} \quad (2.160b)$$

$$V_k = \left( \tilde{R}_k - \tilde{M}'_k \tilde{Q}_k^{-1} \tilde{M}_k \right)^{-1} \quad (2.160c)$$

for  $k \in \mathcal{N}_E$ . The initial state is  $x_{-1} = \tilde{x}_0$  and the other parameters in the state transition equation (2.156b) are

$$A_{-1} = I \quad B_{-1} = I \quad b_{-1} = 0 \quad (2.161a)$$

$$A_k = \tilde{A}_k \quad B_k = \tilde{B}_k \quad b_k = \tilde{b}_k \quad k \in \mathcal{N}_E \quad (2.161b)$$

The parameters in the inequality constraints (2.156c)-(2.156d) are

$$C_{-1} = 0 \quad D_{-1} = 0 \quad c_{-1} = 0 \quad d_{-1} = -1 \quad (2.162a)$$

$$C_k = \tilde{E}_k \quad D_k = \tilde{F}_k \quad c_k = \tilde{h}_k \quad d_k = \tilde{d}_k \quad k \in \mathcal{N}_E \quad (2.162b)$$

$$C_{N+1} = 0 \quad c_{N+1} = 0 \quad d_{N+1} = -1 \quad (2.162c)$$

*Proof.* The basic linear moving horizon estimator is

$$\min_{\{x_k, w_k, v_k\}} \quad \phi = \tilde{l}_{-1}(x_0) + \sum_{k=0}^N \tilde{l}_k(w_k, v_k) \quad (2.163a)$$

$$s.t. \quad x_{k+1} = \tilde{A}'_k x_k + \tilde{B}'_k w_k + \tilde{b}_k \quad k \in \mathcal{N}_E \quad (2.163b)$$

$$y_k = \tilde{C}'_k x_k + \tilde{c}_k + v_k \quad k \in \mathcal{N}_E \quad (2.163c)$$

$$\tilde{E}'_k x_k + \tilde{F}'_k w_k + \tilde{h}_k \geq \tilde{d}_k \quad k \in \mathcal{N}_E \quad (2.163d)$$

in which the stage costs are

$$\tilde{l}_{-1}(x_0) = \frac{1}{2} (x_0 - \tilde{x}_0)' \tilde{P}_0^{-1} (x_0 - \tilde{x}_0) \quad (2.164a)$$

$$\tilde{l}_k(w_k, v_k) = \frac{1}{2} \begin{bmatrix} w_k - \tilde{w}_k \\ v_k - \tilde{v}_k \end{bmatrix}' \begin{bmatrix} \tilde{Q}_k & \tilde{M}_k \\ \tilde{M}_k & \tilde{R}_k \end{bmatrix}^{-1} \begin{bmatrix} w_k - \tilde{w}_k \\ v_k - \tilde{v}_k \end{bmatrix} \quad (2.164b)$$

Define  $x_{-1} = \tilde{x}_0$  and  $w_{-1} = x_0 - \tilde{x}_0$ . This implies

$$\begin{aligned} x_0 &= \tilde{x}_0 + w_{-1} = x_{-1} + w_{-1} \\ &= A'_{-1}x_{-1} + B'_{-1}w_{-1} + b_{-1} \end{aligned} \quad (2.165)$$

in which

$$A_{-1} = I \quad B_{-1} = I \quad b_{-1} = 0 \quad (2.166)$$

Furthermore

$$\begin{aligned} \tilde{l}_{-1}(x_0) &= \frac{1}{2}(x_0 - \tilde{x}_0)' \tilde{P}_0^{-1}(x_0 - \tilde{x}_0) \\ &= \frac{1}{2}w'_{-1} \tilde{P}_0^{-1} w_{-1} = \frac{1}{2}w'_{-1} \Lambda_{-1} w_{-1} \\ &= \frac{1}{2} \|\psi_{-1}(x_{-1}, w_{-1})\|_{\Lambda_{-1}}^2 = l_{-1}(x_{-1}, w_{-1}) \end{aligned} \quad (2.167)$$

in which

$$\psi_{-1}(x_{-1}, w_{-1}) = w_{-1} \quad (2.168)$$

and  $\Lambda_{-1} = \tilde{P}_0^{-1}$ .

At time index  $k = -1$  there is no inequality constraint. To keep a consistent and simple notation, we introduce a dummy inequality constraint which is never active. This inequality constraint may be expressed as

$$C'_{-1}x_{-1} + D'_{-1}w_{-1} + c_{-1} \geq d_{-1} \quad (2.169)$$

in which

$$C_{-1} = 0 \quad D_{-1} = 0 \quad c_{-1} = 0 \quad d_{-1} = -1 \quad (2.170)$$

The measurement equation

$$y_k = \tilde{C}'_k x_k + \tilde{c}_k + v_k \quad k \in \mathcal{N}_E \quad (2.171)$$

may be rearranged to

$$v_k = y_k - \tilde{C}'_k x_k - \tilde{c}_k \quad k \in \mathcal{N}_E \quad (2.172)$$

The stage costs,  $\tilde{l}_k$ , for  $k \in \mathcal{N}_E$  may be expressed as

$$\begin{aligned} \tilde{l}_k(w_k, v_k) &= \frac{1}{2} \begin{bmatrix} w_k - \tilde{w}_k \\ v_k - \tilde{v}_k \end{bmatrix}' \begin{bmatrix} \tilde{Q}_k & \tilde{M}_k \\ \tilde{M}'_k & \tilde{R}_k \end{bmatrix}^{-1} \begin{bmatrix} w_k - \tilde{w}_k \\ v_k - \tilde{v}_k \end{bmatrix} \\ &= \frac{1}{2} \begin{bmatrix} w_k - \tilde{w}_k \\ y_k - \tilde{C}'_k x_k - \tilde{c}_k - \tilde{v}_k \end{bmatrix}' \Lambda_k \begin{bmatrix} w_k - \tilde{w}_k \\ y_k - \tilde{C}'_k x_k - \tilde{c}_k - \tilde{v}_k \end{bmatrix} \\ &= \frac{1}{2} \|\psi_k(x_k, w_k)\|_{\Lambda_k}^2 = l_k(x_k, w_k) \end{aligned} \quad (2.173)$$

in which the residual function,  $\psi_k$ , is

$$\psi_k(x_k, w_k) = \begin{bmatrix} w_k - \tilde{w}_k \\ y_k - \tilde{C}'_k x_k - \tilde{c}_k - \tilde{v}_k \end{bmatrix} \quad (2.174)$$



and the expression for  $\Lambda_k$  is

$$\Lambda_k = \begin{bmatrix} \tilde{Q}_k & \tilde{M}_k \\ \tilde{M}'_k & \tilde{R}_k \end{bmatrix}^{-1} = \begin{bmatrix} W_k & S'_k \\ S_k & V_k \end{bmatrix} \quad (2.175)$$

with  $W_k$ ,  $S_k$ , and  $V_k$  computed according to lemma 2.5.1

$$W_k = \left( \tilde{Q}_k - \tilde{M}_k \tilde{R}_k^{-1} \tilde{M}'_k \right)^{-1} \quad (2.176a)$$

$$S_k = -\tilde{R}_k^{-1} \tilde{M}'_k \left( \tilde{Q}_k - \tilde{M}_k \tilde{R}_k^{-1} \tilde{M}'_k \right)^{-1} \quad (2.176b)$$

$$V_k = \left( \tilde{R}_k - \tilde{M}'_k \tilde{Q}_k^{-1} \tilde{M}_k \right)^{-1} \quad (2.176c)$$

The state transition equation for  $k \in \mathcal{N}_E$  may be expressed as

$$\begin{aligned} x_{k+1} &= \tilde{A}'_k x_k + \tilde{B}'_k w_k + \tilde{b}_k \\ &= A'_k x_k + B'_k w_k + b_k \end{aligned} \quad (2.177)$$

in which

$$A_k = \tilde{A}_k \quad B_k = \tilde{B}_k \quad b_k = \tilde{b}_k \quad (2.178)$$

Similarly, the inequality constraints

$$\tilde{E}'_k x_k + \tilde{F}'_k w_k + \tilde{h}_k \geq \tilde{d}_k \quad (2.179)$$

at time indices  $k \in \mathcal{N}_E$  may be expressed as

$$C'_k x_k + D'_k w_k + c_k \geq d_k \quad (2.180)$$

in which

$$C_k = \tilde{E}_k \quad D_k = \tilde{F}_k \quad c_k = \tilde{h}_k \quad d_k = \tilde{d}_k \quad (2.181)$$

At time index  $k = N + 1$

$$l_{N+1}(x_{N+1}) = 0 = \frac{1}{2} \|\psi_{N+1}(x_{N+1})\|_{\Lambda_{N+1}}^2 \quad (2.182)$$

in which  $\Lambda_{N+1} = 0$  and  $\psi_{N+1}(x_{N+1}) = 0$ . There is no inequality constraints associated with time index  $k = N + 1$ . To keep a consistent and simple notation, we may introduce a dummy constraint which is always inactive

$$C'_{N+1} x_{N+1} + c_{N+1} \geq d_{N+1} \quad (2.183)$$

in which  $C_{N+1} = 0$ ,  $c_{N+1} = 0$ , and  $d_{N+1} = -1$ .  $\square$

Proposition 2.5.8 establishes that the constrained least squares optimal control problem with linear models (2.156) is equivalent with the constrained linear-quadratic optimal control problem (2.184). Furthermore, a procedure for converting the least squares constrained optimal control problem with linear models (2.156) to a constrained linear quadratic optimal control problem (2.184) is provided by proposition 2.5.8.

**Proposition 2.5.8**

The constrained linear least squares optimal control problem (2.156) may be expressed as a constrained linear quadratic optimal control problem

$$\min_{\{x_{k+1}, w_k\}_{k=-1}^N} \phi = \sum_{k=-1}^N f_k(x_k, w_k) + f_{N+1}(x_{N+1}) \quad (2.184a)$$

$$s.t. \quad x_{k+1} = A'_k x_k + B'_k w_k + b_k \quad k \in \{-1\} \cup \mathcal{N}_E \quad (2.184b)$$

$$C'_k x_k + D'_k w_k + c_k \geq d_k \quad k \in \{-1\} \cup \mathcal{N}_E \quad (2.184c)$$

$$C'_{N+1} x_{N+1} + c_{N+1} \geq d_{N+1} \quad (2.184d)$$

The stage costs in the linear quadratic optimal control problem (2.184) are

$$f_k(x_k, w_k) = \frac{1}{2} x'_k Q_k x_k + x'_k M_k w_k + \frac{1}{2} w'_k R_k w_k + q'_k x_k + r'_k w_k + f_k, \quad k \in \{-1\} \cup \mathcal{N}_E \quad (2.185a)$$

$$f_{N+1}(x_{N+1}) = \frac{1}{2} x'_{N+1} P_{N+1} x_{N+1} + p'_{N+1} x_{N+1} + \gamma_{N+1} \quad (2.185b)$$

The stage cost parameters at time index  $k = -1$  are

$$Q_{-1} = 0 \quad M_{-1} = 0 \quad R_{-1} = \tilde{P}_0^{-1} \quad (2.186a)$$

$$q_{-1} = 0 \quad r_{-1} = 0 \quad (2.186b)$$

$$f_{-1} = 0 \quad (2.186c)$$

For the index range  $k \in \mathcal{N}_E = \{0, 1, \dots, N\}$ , define

$$\tilde{y}_k = y_k - \tilde{c}_k - \tilde{v}_k \quad (2.187)$$

such that the parameters of the stage costs in the index range  $k \in \mathcal{N}_E = \{0, 1, \dots, N\}$  may be expressed as

$$Q_k = \tilde{C}_k V_k \tilde{C}'_k \quad M_k = -\tilde{C}_k S_k \quad R_k = W_k \quad (2.188a)$$

$$q_k = \tilde{C}_k (S_k \tilde{w}_k - V_k \tilde{y}_k) \quad r_k = S'_k \tilde{y}_k - W_k \tilde{v}_k \quad (2.188b)$$

$$f_k = \frac{1}{2} \tilde{y}'_k V_k \tilde{y}_k - \tilde{y}'_k S_k \tilde{w}_k + \frac{1}{2} \tilde{w}'_k W_k \tilde{w}_k \quad (2.188c)$$

The parameters associated with the final stage cost,  $f_{N+1}$ , are

$$P_{N+1} = 0 \quad p_{N+1} = 0 \quad \gamma_{N+1} = 0 \quad (2.189)$$

The initial state,  $x_{-1}$ , and the parameters of the constraints (2.184b)-(2.184d) are as in proposition 2.5.7.

*Proof.* To convert the constrained linear least squares optimal control problem (2.156) to the constrained linear quadratic optimal control problem (2.184), we must show that the least squares stage costs (2.157) may be expressed as (2.185).

At time index  $k = -1$ , the stage costs of (2.156) may be expressed as

$$\begin{aligned} l_{-1}(w_{-1}, x_{-1}) &= \frac{1}{2} \|\psi_{-1}(w_{-1}, x_{-1})\|_{\Lambda_{-1}}^2 \\ &= \frac{1}{2} w_{-1} \tilde{P}_0^{-1} w_{-1} \\ &= \frac{1}{2} x'_{-1} Q_{-1} x_{-1} + x'_{-1} M_{-1} w_{-1} + \frac{1}{2} w'_{-1} R_{-1} w_{-1} \\ &\quad + q'_{-1} x_{-1} + r'_{-1} w_{-1} + f_{-1} \end{aligned} \quad (2.190)$$

in which

$$Q_{-1} = 0 \quad M_{-1} = 0 \quad R_{-1} = \tilde{P}_0^{-1} \quad (2.191a)$$

$$q_{-1} = 0 \quad r_{-1} = 0 \quad (2.191b)$$

$$f_{-1} = 0 \quad (2.191c)$$

For notational convenience define

$$\tilde{y}_k = y_k - \tilde{c}_k - \tilde{v}_k \quad k \in \mathcal{N}_E \quad (2.192)$$

The stage costs of (2.156) at time index  $k \in \mathcal{N}_E$  may be expressed as

$$\begin{aligned} l_k(x_k, w_k) &= \frac{1}{2} \|\psi_k(x_k, w_k)\|_{\Lambda_k}^2 \\ &= \frac{1}{2} \begin{bmatrix} w_k - \tilde{w}_k \\ \tilde{y}_k - \tilde{C}'_k x_k \end{bmatrix}' \begin{bmatrix} W_k & S'_k \\ S_k & V_k \end{bmatrix} \begin{bmatrix} w_k - \tilde{w}_k \\ \tilde{y}_k - \tilde{C}'_k x_k \end{bmatrix} \\ &= \frac{1}{2} (w_k - \tilde{w}_k)' W_k (w_k - \tilde{w}_k) + (\tilde{y}_k - \tilde{C}'_k x_k)' S_k (w_k - \tilde{w}_k) \\ &\quad + \frac{1}{2} (\tilde{y}_k - \tilde{C}'_k x_k)' V_k (\tilde{y}_k - \tilde{C}'_k x_k) \\ &= \frac{1}{2} x'_k \tilde{C}_k V_k \tilde{C}'_k x_k + x'_k (-\tilde{C}_k S_k) w_k + \frac{1}{2} w'_k W_k w_k \\ &\quad + (\tilde{w}'_k S'_k - \tilde{y}'_k V_k) \tilde{C}'_k x_k + (-\tilde{w}'_k W_k + \tilde{y}'_k S_k) w_k \\ &\quad + \left( \frac{1}{2} \tilde{w}'_k W_k \tilde{w}_k - \tilde{y}'_k S_k \tilde{w}_k + \frac{1}{2} \tilde{y}'_k V_k \tilde{y}_k \right) \\ &= \frac{1}{2} x'_k Q_k x_k + x'_k M_k w_k + \frac{1}{2} w'_k R_k w_k + q'_k x_k + r'_k w_k + f_k \end{aligned} \quad (2.193)$$

in which we have introduced and defined the following quantities

$$Q_k = \tilde{C}_k W_k \tilde{C}'_k \quad M_k = -\tilde{C}_k S_k \quad R_k = W_k \quad (2.194a)$$

$$q_k = \tilde{C}_k (S_k \tilde{w}_k - V_k \tilde{y}_k) \quad r_k = S'_k \tilde{y}_k - W_k \tilde{w}_k \quad (2.194b)$$

$$f_k = \frac{1}{2} \tilde{w}'_k W_k \tilde{w}_k - \tilde{y}'_k S_k \tilde{w}_k + \frac{1}{2} \tilde{y}'_k V_k \tilde{y}_k \quad (2.194c)$$

The stage cost of (2.156) at time index  $k = N + 1$  is

$$\begin{aligned} l_{N+1}(x_{N+1}) &= \frac{1}{2} \|\psi_{N+1}(x_{N+1})\|_{\Lambda_{N+1}}^2 = 0 \\ &= \frac{1}{2} x'_{N+1} P_{N+1} x_{N+1} + p'_{N+1} x_{N+1} + \gamma_{N+1} \end{aligned} \quad (2.195)$$

in which

$$P_{N+1} = 0 \quad p_{N+1} = 0 \quad \gamma_{N+1} = 0 \quad (2.196)$$

□

Once again, by this section the central role of the constrained linear-quadratic optimal control problem has been demonstrated. By propositions 2.5.6, 2.5.7, and 2.5.8 a procedure for converting the augmented moving horizon estimator with a linear model (2.139) to a constrained linear-quadratic optimal control problem (2.184) has been established. The implication of this result is that an efficient procedure for solving the constrained linear-quadratic optimal control problem also may be used to efficiently solve the augmented moving horizon estimator with a linear model.

## 2.6 Moving Horizon Control

Based on the information available, the moving horizon estimator computes an estimate of the current states. These estimated states may consist of both the original states and some augmented states introduced to ensure offset free control. The state vector contains the minimum information needed to predict the future evolution of systems governed by either ordinary differential equation systems or ordinary difference equation systems.

Given the current state, the moving horizon controller selects a sequence of inputs  $\{u_k^*\}$  such that some performance measure is optimized. The sequence of inputs  $\{u_k^*\}$  is obtained by solving an open-loop optimal control problem. The forecast of the performance measure is based on the model as well as the current state estimate. Only the first input,  $u_0^*$ , in the sequence of inputs  $\{u_k^*\}$  computed by the moving horizon controller is injected to the process. Feedback is achieved by repeating the process as new measurements become available.

### 2.6.1 Nonlinear Moving Horizon Control

The general nonlinear moving horizon controller is closely related to the nonlinear optimal control problem. However, moving horizon controllers are typically formulated on an infinite horizon and may involve variables of the type  $\Delta u_k = u_k - u_{k-1}$ . The infinite horizon formulation is made practical by solving some finite horizon problem that approximates the infinite horizon problem. Movement variables,  $\Delta u_k$ , are eliminated from the formulation by augmenting the state vector with the previous input  $u_{k-1}$ .

To illustrate the principles in converting a moving horizon control problem to a nonlinear optimal control problem (2.43), consider the infinite-horizon moving horizon control problem

$$\min_{\{w_{k+1}, u_k\}_{k=0}^{\infty}} \phi = \sum_{k=0}^{\infty} \tilde{f}_k(w_k, u_k, \Delta u_k, z_k) \quad (2.197a)$$

$$s.t. \quad w_{k+1} = \tilde{g}_k(w_k, u_k, v_k) \quad (2.197b)$$

$$\tilde{h}_k(w_k, u_k, \Delta u_k) \geq d_k \quad (2.197c)$$

in which the state vectors,  $\{w_{k+1}\}_{k=0}^{\infty}$ , and the input vectors,  $\{u_k\}_{k=0}^{\infty}$ , are the decision variables. The initial state,  $w_0$ , is a parameter. Further, the exogenous inputs  $\{z_k\}_{k=0}^{\infty}$  and  $\{v_k\}_{k=0}^{\infty}$  are parameters in (2.197). The sequence  $\{z_k\}_{k=0}^{\infty}$  may represent a specified setpoint trajectory, but it may also for instance be prices associated with the objective function. As an example of the latter, for a power consuming process,  $\{z_k\}_{k=0}^{\infty}$ , may represent the price of electricity during the day or week.  $\{v_k\}_{k=0}^{\infty}$  represents known disturbances. The sequence,  $\{v_k\}_{k=0}^{\infty}$  may be used to represent a scheduled change in the raw materials fed to the system, i.e. a switch of crude oil to a refinery. For

some systems, in which the inventory of some product is modeled,  $\{v_k\}_{k=0}^{\infty}$  may represent the forecasted demand for that particular product. The stage costs in the objective function of (2.197) has a very general structure and may be used to represent deviations from a target as well as general economic cost functions. When an economic cost function is employed, the optimal control problem (2.197) is sometimes referred to as optimizing control.

The infinite horizon nonlinear moving horizon control problem is a mathematical abstraction which cannot in general be solved numerically. Nominally the infinite horizon nonlinear moving horizon controller will predict the actual closed-loop trajectory perfectly and as a consequence be stabilizing provided a stabilizing controller exists. Any practically implementable moving horizon controller should mimic the infinite horizon nonlinear moving horizon controller due to the nominal stability, excellent performance, intuitive tuning and design, and theoretical correctness of the infinite horizon nonlinear moving horizon controller. The first issue in constructing a numerically tractable approximation to the infinite horizon moving horizon controller concerns representation of (2.197) by a finite data set. In doing so we assume that (2.197) has a finite parameterization. Let the functions  $(\tilde{f}_k, \tilde{g}_k, \tilde{h}_k)$  that constitute the infinite-horizon moving horizon control problem (2.197) be represented by the structure

$$\begin{aligned} \mathcal{C} &= \{\mathcal{M}_k\}_{k=0}^{\infty} \\ &= \{\mathcal{M}_0, \mathcal{M}_1, \dots, \mathcal{M}_{N-1}, \mathcal{M}, \mathcal{M}, \dots\} \end{aligned} \quad (2.198)$$

in which  $\mathcal{M}_k = \{\tilde{f}_k, \tilde{g}_k, \tilde{h}_k\}$  and  $\mathcal{M} = \{\tilde{f}, \tilde{g}, \tilde{h}\}$ . The exogeneous parameters of (2.197) are assumed to have the parameterization

$$\{z_k\}_{k=0}^{\infty} = \{z_0, z_1, \dots, z_{N-1}, z, z, \dots\} \quad (2.199a)$$

$$\{v_k\}_{k=0}^{\infty} = \{v_0, v_1, \dots, v_{N-1}, v, v, \dots\} \quad (2.199b)$$

$$\{d_k\}_{k=0}^{\infty} = \{d_0, d_1, \dots, d_{N-1}, d, d, \dots\} \quad (2.199c)$$

The second issue in solving (2.197) concerns the construction of an approximate solution given the finite data set representing the problem. Typically, the approximate solution of (2.197) is computed by assuming that the optimal solution of (2.197) converges toward an optimal steady state (c.f. Rawlings, 2000). The optimal steady state associated with (2.197) is computed as the solution to the mathematical program

$$\min_{w_s, u_s} \quad \bar{f}(w_s, u_s, z) \quad (2.200a)$$

$$s.t. \quad w_s = \tilde{g}(w_s, u_s, v) \quad (2.200b)$$

$$\tilde{h}(w_s, u_s, 0) \geq d \quad (2.200c)$$

in which  $\bar{f}$  may be  $\bar{f}(w_s, u_s, z) = \tilde{f}(w_s, u_s, 0, z)$  or some modification of this expression. When the problem (2.197) is a discretization of an underlying

continuous-time problem, the optimal steady state may be found using the continuous steady state condition rather than (2.200b).

The solution of the dynamic part of the infinite horizon moving horizon control problem is facilitated by construction of an equivalent nonlinear optimal control problem. The infinite horizon moving horizon control problem (2.197) may be transformed to an infinite horizon nonlinear optimal control problem by state augmentation. This transformation is described in the following proposition.

**Proposition 2.6.1**

Let the augmented state  $x_k$  be

$$x_k = \begin{bmatrix} w_k \\ u_{k-1} \end{bmatrix} \quad (2.201)$$

and let  $\Delta u_k = u_k - u_{k-1}$ .

Then the infinite-horizon moving horizon control problem (2.197) is equivalent to the infinite-horizon constrained optimal control problem

$$\min_{\{x_{k+1}, u_k\}} \phi = \sum_{k=0}^{\infty} f_k(x_k, u_k) \quad (2.202a)$$

$$s.t. \quad x_{k+1} = g_k(x_k, u_k, v_k) \quad (2.202b)$$

$$h_k(x_k, u_k) \geq d_k \quad (2.202c)$$

in which

$$f_k(x_k, u_k) = f_k(x_k, u_k, z_k) = \tilde{f}_k(w_k, u_k, \Delta u_k, z_k) \quad (2.203a)$$

$$g_k(x_k, u_k) = g_k(x_k, u_k, v_k) = \begin{bmatrix} \tilde{g}_k(w_k, u_k, v_k) \\ u_k \end{bmatrix} \quad (2.203b)$$

$$h_k(x_k, u_k) = \tilde{h}_k(w_k, u_k, \Delta u_k) \quad (2.203c)$$

*Proof.* Consider the infinite horizon moving horizon control problem (2.197). Define the augmented state  $x_k$  as

$$x_k = \begin{bmatrix} w_k \\ u_{k-1} \end{bmatrix} \quad (2.204)$$

and observe that

$$w_k = \begin{bmatrix} I \\ 0 \end{bmatrix}' \begin{bmatrix} w_k \\ u_{k-1} \end{bmatrix} = \Phi' x_k \quad (2.205a)$$

$$\Delta u_k = u_k - u_{k-1} = \begin{bmatrix} 0 \\ -I \end{bmatrix}' \begin{bmatrix} w_k \\ u_{k-1} \end{bmatrix} + u_k = \Gamma' x_k + u_k \quad (2.205b)$$

Then

$$\tilde{f}_k(w_k, u_k, \Delta u_k, z_k) = \tilde{f}_k(\Phi' x_k, u_k, \Gamma' x_k + u_k, z_k) = f_k(x_k, u_k) \quad (2.206)$$

In the expression,  $f_k(x_k, u_k)$ ,  $z_k$  is not included as an argument even though the function  $f_k(x_k, u_k)$  depends on  $z_k$ . The reason is that  $z_k$  is a parameter

and not a decision variable in the infinite horizon nonlinear moving horizon control problem (2.197). The dependence of  $f_k(x_k, u_k)$  on  $z_k$  is incorporated by the definition of  $f_k(x_k, u_k)$ .

The augmented state transition is given by

$$x_{k+1} = \begin{bmatrix} w_{k+1} \\ u_k \end{bmatrix} = \begin{bmatrix} \tilde{g}_k(x_k, u_k, v_k) \\ u_k \end{bmatrix} = g_k(x_k, u_k, v_k) = g_k(x_k, u_k) \quad (2.207)$$

$v_k$  is not included as an argument in the function  $g_k(x_k, u_k)$  as  $v_k$  is a parameter and not a decision variable in the infinite horizon nonlinear optimal control problem (2.197).

The inequality constraint function may be expressed as

$$\tilde{h}_k(w_k, u_k, \Delta u_k) = \tilde{h}_k(\Phi' x_k, u_k, \Gamma' x_k + u_k) = h_k(x_k, u_k) \quad (2.208)$$

which implies

$$h_k(x_k, u_k) \geq d_k \quad (2.209)$$

These expressions for the stage costs, the state transition function, and the inequality constraints demonstrate the procedure stated in the proposition for converting the infinite horizon nonlinear moving horizon control problem (2.197) to an infinite horizon optimal control problem (2.202).  $\square$

The infinite horizon optimal control formulation (2.202) of the general nonlinear moving horizon control problem (2.197) cannot in general be solved as it has an infinite number of decision variables and in general no analytical solution is available. Therefore, a finite-dimensional approximation

$$\min_{\{x_{k+1}, u_k\}_{k=0}^{N-1}} \phi = \sum_{k=0}^{N-1} f_k(x_k, u_k) + F_N(x_N) \quad (2.210a)$$

$$s.t. \quad x_{k+1} = g_k(x_k, u_k) \quad k \in \mathcal{N} \quad (2.210b)$$

$$h_k(x_k, u_k) \geq d_k \quad k \in \mathcal{N} \quad (2.210c)$$

$$H_N(x_N) \geq \bar{d}_N \quad (2.210d)$$

of the infinite-dimensional problem (2.202) is solved in practical moving horizon controllers. The selection of the approximating functions,  $F_N(x_N)$  and  $H_N(x_N)$ , is outside the scope of this paper. Detailed discussions and overviews can be found in Chen and Allgöwer (1998), Gilbert and Tan (1991), Mayne *et al.* (2000), and De Nicolao *et al.* (2000). For the present purpose, it suffices to observe that the terminal stage cost is often selected to have the form

$$F_N(x_N) = \frac{1}{2}(x_N - x_s)' P_N (x_N - x_s) \quad (2.211)$$

in which  $x_s = [w'_s, u'_s]'$  is the optimal solution of the target problem (2.200).

In this discussion, it has been argued that the practical approximate solution of the general nonlinear moving horizon controller (2.197) may be obtained

by solving a target problem (2.200) and a finite-dimensional nonlinear optimal control problem (2.210). The target problem (2.200) is an instance of a general nonlinear program and may for instance be solved by standard SQP algorithms. The finite-dimensional nonlinear optimal control problem (2.202) is an instance of the nonlinear optimal control problem (2.43) and may be solved applying an SQP algorithm. The quadratic program in an SQP algorithm for the nonlinear optimal control problem has the structure of a constrained linear quadratic optimal control problem (2.30) provided the Hessian matrix in this quadratic program is computed as a structured BFGS update of the Hessian matrix of (2.43).

In conclusion, the computationally expensive part of the discrete-time general moving horizon control is computation of the search directions in the SQP algorithm. These directions are obtained by solving constrained linear quadratic optimal control problems (2.30).

### 2.6.2 Least-Squares Nonlinear Moving Horizon Control

The nonlinear least-squares moving horizon control problem is the following infinite-horizon optimal control problem

$$\min_{\{w_{k+1}, u_k, y_k\}} \phi = \sum_{k=0}^{\infty} \tilde{l}_k(y_k, \Delta u_k, z_k) \quad (2.212a)$$

$$s.t. \quad w_{k+1} = \tilde{g}_k(w_k, u_k, v_k) \quad (2.212b)$$

$$\tilde{h}_k(w_k, u_k, \Delta u_k) \geq d_k \quad (2.212c)$$

$$y_k = \tilde{\varphi}_k(w_k, u_k) \quad (2.212d)$$

with stage costs of the least-squares type

$$\tilde{l}_k(y_k, \Delta u_k, z_k) = \frac{1}{2}(y_k - z_k)' \tilde{\Lambda}_k (y_k - z_k) + \frac{1}{2} \Delta u_k' \tilde{S}_k \Delta u_k \quad (2.213)$$

The mathematical program (2.212) is assumed to have the following finite parameterization

$$\begin{aligned} \mathcal{C} &= \{\mathcal{M}_k\}_{k=0}^{\infty} \\ &= \{\mathcal{M}_0, \mathcal{M}_1, \dots, \mathcal{M}_{N-1}, \mathcal{M}, \mathcal{M}, \dots\} \end{aligned} \quad (2.214a)$$

$$\{z_k\}_{k=0}^{\infty} = \{z_0, z_1, \dots, z_{N-1}, z, z, \dots\} \quad (2.214b)$$

$$\{v_k\}_{k=0}^{\infty} = \{v_0, v_1, \dots, v_{N-1}, v, v, \dots\} \quad (2.214c)$$

in which  $\mathcal{M}_k = \{\tilde{\Lambda}_k, \tilde{S}_k, \tilde{g}_k, \tilde{h}_k, d_k, \tilde{\varphi}_k\}$  and  $\mathcal{M} = \{\tilde{\Lambda}, \tilde{S}, \tilde{g}, \tilde{h}, d, \tilde{\varphi}\}$ . The practical nonlinear least-squares moving horizon controller obtains an *approximate* solution to (2.212) by solving a target problem and a regulation problem. The



target problem determines the optimal steady-state given the exogeneous inputs  $(z, v)$

$$\min_{w_s, u_s, y_s} \quad \bar{l}(y_s, u_s, z) \quad (2.215a)$$

$$s.t. \quad w_s = \tilde{g}(w_s, u_s, v) \quad (2.215b)$$

$$\tilde{h}(w_s, u_s, 0) \geq d \quad (2.215c)$$

$$y_s = \tilde{\varphi}(w_s, u_s) \quad (2.215d)$$

The objective function of the target problem is

$$\bar{l}(y_s, u_s, z) = \frac{1}{2}(y_s - z)' \tilde{\Lambda}(y_s - z) + \frac{1}{2}(u_s - \bar{u})' R_s (u_s - \bar{u}) \quad (2.216)$$

in which  $R_s$  is selected such that the second term does not interfere with the first term. By this choice,  $u_s$  is selected such that it is close to  $\bar{u}$  if the process has excess degrees of freedom.

To solve the regulation problem, the nonlinear least-squares moving horizon controller is converted to an optimal control problem. The first step in doing so is to eliminate  $\Delta u_k$  by augmentation of the state. This conversion is stated in the following proposition.

**Proposition 2.6.2**

Let

$$x_k = \begin{bmatrix} w_k \\ u_{k-1} \end{bmatrix} \quad \Delta u_k = u_k - u_{k-1} \quad (2.217)$$

The the nonlinear least-squares moving horizon control problem (2.212) may be stated as the following infinite-horizon optimal control problem

$$\min_{\{x_{k+1}, u_k\}} \quad \phi = \sum_{k=0}^{\infty} l_k(x_k, u_k) \quad (2.218a)$$

$$s.t. \quad x_{k+1} = g_k(x_k, u_k) \quad (2.218b)$$

$$h_k(x_k, u_k) \geq d_k \quad (2.218c)$$

with least-squares stage costs

$$l_k(x_k, u_k) = \frac{1}{2} \|\psi_k(x_k, u_k)\|_{\Lambda_k}^2 \quad (2.219)$$

The residual functions  $\psi_k$  are

$$\begin{aligned} \psi_k(x_k, u_k) &= \begin{bmatrix} \varphi_k(x_k, u_k) - z_k \\ \Gamma' x_k + u_k \end{bmatrix} \\ &= \begin{bmatrix} \tilde{\varphi}_k(w_k, u_k) - z_k \\ \Delta u_k \end{bmatrix} \end{aligned} \quad (2.220)$$

in which

$$\varphi_k(x_k, u_k) = \tilde{\varphi}_k(w_k, u_k) \quad (2.221)$$

and

$$\Gamma' = \begin{bmatrix} 0 & -I \end{bmatrix} \quad (2.222)$$

The weight matrices  $\Lambda_k$  are

$$\Lambda_k = \begin{bmatrix} \tilde{\Lambda}_k & 0 \\ 0 & \tilde{S}_k \end{bmatrix} \quad (2.223)$$

and the constraint functions  $g_k$  and  $h_k$  are

$$g_k(x_k, u_k) = g_k(x_k, u_k, v_k) = \begin{bmatrix} \tilde{g}_k(w_k, u_k, v_k) \\ u_k \end{bmatrix} \quad (2.224a)$$

$$h_k(x_k, u_k) = \tilde{h}_k(w_k, u_k, \Delta u_k) \quad (2.224b)$$

*Proof.* Consider the nonlinear least-squares moving horizon control problem (2.212). Define the augmented state

$$x_{k+1} = \begin{bmatrix} w_k \\ u_{k-1} \end{bmatrix} \quad (2.225)$$

Then

$$w_k = \begin{bmatrix} I \\ 0 \end{bmatrix}' \begin{bmatrix} w_k \\ u_{k-1} \end{bmatrix} = \Phi' x_k \quad \Phi = \begin{bmatrix} I \\ 0 \end{bmatrix} \quad (2.226a)$$

$$\Delta u_k = u_k - u_{k-1} = \begin{bmatrix} 0 \\ -I \end{bmatrix}' \begin{bmatrix} w_k \\ u_{k-1} \end{bmatrix} + u_k = \Gamma' x_k + u_k \quad \Gamma = \begin{bmatrix} 0 \\ -I \end{bmatrix} \quad (2.226b)$$

and

$$y_k = \tilde{\varphi}_k(w_k, u_k) = \tilde{\varphi}_k(\Phi' x_k, u_k) = \varphi_k(x_k, u_k) \quad (2.227)$$

These expressions may be used to establish that the stage costs are of the least-squares type

$$\begin{aligned} \tilde{l}_k(y_k, \Delta u_k, z_k) &= \frac{1}{2}(y_k - z_k)' \tilde{\Lambda}_k (y_k - z_k) + \frac{1}{2} \Delta u_k' \tilde{S}_k \Delta u_k \\ &= \frac{1}{2}(\varphi_k(x_k, u_k) - z_k)' \tilde{\Lambda}_k (\varphi_k(x_k, u_k) - z_k) \\ &\quad + \frac{1}{2}(\Gamma' x_k + u_k)' \tilde{S}_k (\Gamma' x_k + u_k) \\ &= \frac{1}{2} \begin{bmatrix} \varphi_k(x_k, u_k) - z_k \\ \Gamma' x_k + u_k \end{bmatrix}' \begin{bmatrix} \tilde{\Lambda}_k & 0 \\ 0 & \tilde{S}_k \end{bmatrix} \begin{bmatrix} \varphi_k(x_k, u_k) - z_k \\ \Gamma' x_k + u_k \end{bmatrix} \\ &= \frac{1}{2} \psi_k(x_k, u_k)' \Lambda_k \psi_k(x_k, u_k) \\ &= \frac{1}{2} \|\psi_k(x_k, u_k)\|_{\Lambda_k}^2 = l_k(x_k, u_k) \end{aligned} \quad (2.228)$$

in which we have introduced the residual function

$$\psi_k(x_k, u_k) = \begin{bmatrix} \varphi_k(x_k, u_k) - z_k \\ \Gamma' x_k + u_k \end{bmatrix} = \begin{bmatrix} \tilde{\varphi}_k(w_k, u_k) - z_k \\ \Delta u_k \end{bmatrix} \quad (2.229)$$

and the weight matrix

$$\Lambda_k = \begin{bmatrix} \tilde{\Lambda}_k & 0 \\ 0 & \tilde{S}_k \end{bmatrix} \quad (2.230)$$

The state transition equation for the augmented state,  $x_k$ , is

$$\begin{aligned} x_{k+1} &= \begin{bmatrix} w_{k+1} \\ u_k \end{bmatrix} = \begin{bmatrix} \tilde{g}_k(w_k, u_k, v_k) \\ u_k \end{bmatrix} = \begin{bmatrix} \tilde{g}_k(\Phi' x_k, u_k, v_k) \\ u_k \end{bmatrix} \\ &= g_k(x_k, u_k, v_k) = g_k(x_k, u_k) \end{aligned} \quad (2.231)$$

and the inequality constraints may be expressed as

$$d_k \leq \tilde{h}_k(w_k, u_k, \Delta u_k) = \tilde{h}_k(\Phi' x_k, u_k, \Gamma' x_k + u_k) = h_k(x_k, u_k) \quad (2.232)$$

Consequently, the nonlinear least-squares moving horizon controller (2.212) may be expressed as (2.218).  $\square$

An approximate solution of (2.218) is constructed by computing the optimal solution to a finite-horizon approximation of (2.218)

$$\min_{\{x_{k+1}, u_k\}_{k=0}^{N-1}} \phi = \sum_{k=0}^{N-1} l_k(x_k, u_k) + L_N(x_N) \quad (2.233a)$$

$$s.t. \quad x_{k+1} = g_k(x_k, u_k) \quad k = 0, 1, \dots, N-1 \quad (2.233b)$$

$$h_k(x_k, u_k) \geq d_k \quad k = 0, 1, \dots, N-1 \quad (2.233c)$$

$$H_N(x_N) \geq \bar{d}_N \quad (2.233d)$$

While the actual selection of the functions  $H_N$  and  $L_N$  is outside the scope of this paper, it is important to note that  $L_N$  is of the least-squares type

$$\begin{aligned} L_N(x_N) &= \frac{1}{2}(x_N - x_s)' P (x_N - x_s) \\ &= \frac{1}{2} \|x_N - x_s\|_P^2 \end{aligned} \quad (2.234)$$

and connected to the target problem by  $x_s = [w'_s, u'_s]'$  as  $(w_s, u_s)$  is the optimal solution of the target problem (2.215). The finite horizon approximation (2.233) of (2.218) is a constrained optimal control problem with least squares stage costs. Solving this problem with an SQP algorithm involves solution of a quadratic program with the structure of a constrained linear optimal control problem. Due the least squares structure of the stage costs in (2.233), the Hessian matrix of the quadratic program may be obtained using the Gauss-Newton approximation, the structured BFGS update, or a hybrid method. The following proposition specializes the quadratic program obtained in an SQP algorithm using the Gauss-Newton approximation to the regulator problem (2.233) in the moving horizon controller (2.212) with least squares stage costs.

### Proposition 2.6.3

Consider the nonlinear least-squares moving horizon control problem (2.212). Let the regulator part of the approximate solution to (2.212) be obtained as the solution to (2.233). Let this solution be generated by an SQP algorithm and let  $\{w_{k+1}^0, u_k^0\}_{k=0}^{N-1}$  be the trajectory at an iteration in the SQP algorithm. Let

$$x_k = \begin{bmatrix} w_k \\ u_{k-1} \end{bmatrix} \quad \hat{x}_k = \begin{bmatrix} \hat{w}_k \\ \hat{u}_{k-1} \end{bmatrix} \quad (2.235)$$

and  $\Delta u_k = u_k - u_{k-1}$ . Let

$$x_0^0 = \begin{bmatrix} w_0^0 \\ u_{-1}^0 \end{bmatrix} = \begin{bmatrix} w_0 \\ u_{-1} \end{bmatrix} = x_0 \quad x_s = \begin{bmatrix} w_s \\ u_s \end{bmatrix} \quad (2.236)$$

in which  $x_0$  is a given parameter and  $x_s$  is part of the solution of the target problem (2.215).

Then the search direction  $\{\hat{x}_{k+1}, \hat{u}_k\}_{k=0}^{N-1}$  of the SQP algorithm is obtained as the solution of the constrained linear quadratic optimal control problem

$$\min_{\{\hat{x}_{k+1}, \hat{u}_k\}} \phi = \sum_{k=0}^{N-1} l_k(\hat{x}_k, \hat{u}_k) + l_N(\hat{x}_N) \quad (2.237a)$$

$$s.t. \quad \hat{x}_{k+1} = A'_k \hat{x}_k + B'_k \hat{u}_k + b_k \quad (2.237b)$$

$$C'_k \hat{x}_k + D'_k \hat{u}_k + c_k \geq d_k \quad (2.237c)$$

$$C'_N \hat{x}_N + c_N \geq d_N \quad (2.237d)$$

in which the stage costs are

$$l_k(\hat{x}_k, \hat{u}_k) = \frac{1}{2} (\hat{x}'_k Q_k \hat{x}_k + 2\hat{x}'_k M_k \hat{u}_k + \hat{u}'_k R_k \hat{u}_k) + q'_k \hat{x}_k + r'_k \hat{u}_k + f_k \quad (2.238a)$$

$$l_N(\hat{x}_N) = \frac{1}{2} \hat{x}'_N P_N \hat{x}_N + p'_N \hat{x}_N + \gamma_N \quad (2.238b)$$

and the initial state  $\hat{x}_0$  is a parameter with the value  $\hat{x}_0 = 0$ .

For  $k = 0, 1, \dots, N-1$  the auxiliary parameters  $y_k^0$  and  $\Delta u_k^0$  defined by

$$y_k^0 = \tilde{\varphi}_k(w_k^0, u_k^0) \quad (2.239a)$$

$$\Delta u_k^0 = u_k^0 - u_{k-1}^0 \quad (2.239b)$$

are used for computing the weight matrices of the stage costs in (2.238a)

$$Q_k = \begin{bmatrix} \nabla_{w_k} \tilde{\varphi}_k(w_k^0, u_k^0) \tilde{\Lambda}_k \nabla_{w_k} \tilde{\varphi}_k(w_k^0, u_k^0)' & 0 \\ 0 & \tilde{S}_k \end{bmatrix} \quad (2.240a)$$

$$M_k = \begin{bmatrix} \nabla_{w_k} \tilde{\varphi}_k(w_k^0, u_k^0) \tilde{\Lambda}_k \nabla_{u_k} \tilde{\varphi}_k(w_k^0, u_k^0)' \\ -\tilde{S}_k \end{bmatrix} \quad (2.240b)$$

$$R_k = \nabla_{u_k} \tilde{\varphi}_k(w_k^0, u_k^0) \tilde{\Lambda}_k \nabla_{u_k} \tilde{\varphi}_k(w_k^0, u_k^0)' + \tilde{S}_k \quad (2.240c)$$

$$q_k = \begin{bmatrix} \nabla_{w_k} \tilde{\varphi}_k(w_k^0, u_k^0) \tilde{\Lambda}_k (y_k^0 - z_k) \\ -\tilde{S}_k \Delta u_k^0 \end{bmatrix} \quad (2.240d)$$

$$r_k = \nabla_{u_k} \tilde{\varphi}_k(w_k^0, u_k^0) \tilde{\Lambda}_k (y_k^0 - z_k) + \tilde{S}_k \Delta u_k^0 \quad (2.240e)$$

$$f_k = \frac{1}{2} (y_k^0 - z_k)' \tilde{\Lambda}_k (y_k^0 - z_k) + \frac{1}{2} \Delta u_k^0 \tilde{S}_k \Delta u_k^0 \quad (2.240f)$$

as well as the equality constraint parameters

$$A_k = \begin{bmatrix} \nabla_{w_k} \tilde{g}_k(w_k^0, u_k^0, v_k) & 0 \\ 0 & 0 \end{bmatrix} \quad (2.241a)$$

$$B_k = [\nabla_{u_k} \tilde{g}_k(w_k^0, u_k^0, v_k) \quad I] \quad (2.241b)$$

$$b_k = \begin{bmatrix} \tilde{g}_k(w_k^0, u_k^0, v_k) - w_{k+1}^0 \\ 0 \end{bmatrix} \quad (2.241c)$$

and the inequality constraint parameters

$$C_k = \begin{bmatrix} \nabla_{w_k} \tilde{h}_k(w_k^0, u_k^0, \Delta u_k^0) \\ -\nabla_{\Delta u_k} \tilde{h}_k(w_k^0, u_k^0, \Delta u_k^0) \end{bmatrix} \quad (2.242a)$$

$$D_k = \nabla_{u_k} \tilde{h}_k(w_k^0, u_k^0, \Delta u_k^0) + \nabla_{\Delta u_k} \tilde{h}_k(w_k^0, u_k^0, \Delta u_k^0) \quad (2.242b)$$

$$c_k = \tilde{h}_k(w_k^0, u_k^0, \Delta u_k^0) \quad (2.242c)$$

The weight matrices in the terminal cost function (2.238b) are

$$P_N = P \quad p_N = -Px_s \quad \gamma_N = \frac{1}{2}x_s'Px_s \quad (2.243)$$

and the parameters of the inequality constraint at time index  $k = N$  are

$$C_N = \nabla_{x_N} H_N(x_N^0) \quad (2.244a)$$

$$c_N = H_N(x_N^0) \quad (2.244b)$$

*Proof.* Note that the augmented state is given by

$$x_k = \begin{bmatrix} w_k \\ u_{k-1} \end{bmatrix} \quad (2.245)$$

and that we have the relations

$$w_k = \begin{bmatrix} I \\ 0 \end{bmatrix}' \begin{bmatrix} w_k \\ u_{k-1} \end{bmatrix} = \Phi' x_k \quad (2.246a)$$

$$\Delta u_k = u_k - u_{k-1} = \begin{bmatrix} 0 \\ -I \end{bmatrix}' \begin{bmatrix} w_k \\ u_{k-1} \end{bmatrix} + u_k = \Gamma' x_k + u_k \quad (2.246b)$$

Let the search direction be denoted by  $\{\hat{x}_{k+1}, \hat{u}_k\}_{k=0}^{N-1} = \{x_{k+1} - x_{k+1}^0, u_k - u_k^0\}_{k=0}^{N-1}$ . Note that  $\hat{x}_0 = x_0 - x_0^0 = 0$  as  $x_0$  is a parameter and not a decision variable in the nonlinear least-squares moving horizon control problem. The constraints of the QP used for computing the search direction in an SQP algorithm are given by linearization of the nonlinear constraints around the current trajectory,  $\{x_{k+1}^0, u_k^0\}_{k=0}^{N-1}$ .

Linearization of the state transition equation

$$x_{k+1} = g_k(x_k, u_k) \quad (2.247)$$

yields

$$x_{k+1}^0 + \hat{x}_{k+1} = g_k(x_k^0, u_k^0) + \nabla_{x_k} g_k(x_k^0, u_k^0)' \hat{x}_k + \nabla_{u_k} g_k(x_k^0, u_k^0) \hat{u}_k \quad (2.248)$$

which may be rearranged to

$$\hat{x}_{k+1} = A_k' \hat{x}_k + B_k' \hat{u}_k + b_k \quad (2.249)$$

with

$$\begin{aligned} A_k &= \nabla_{x_k} g_k(x_k^0, u_k^0) = \nabla_{x_k} \begin{bmatrix} \tilde{g}_k(\Phi' x_k^0, u_k^0, v_k) \\ u_k \end{bmatrix} \\ &= \begin{bmatrix} \nabla_{w_k} \tilde{g}_k(w_k^0, u_k^0, v_k) & 0 \\ 0 & 0 \end{bmatrix} \end{aligned} \quad (2.250a)$$

$$\begin{aligned} B_k &= \nabla_{u_k} g_k(x_k^0, u_k^0) = \nabla_{u_k} \begin{bmatrix} \tilde{g}_k(w_k^0, u_k^0, v_k) \\ u_k \end{bmatrix} \\ &= \begin{bmatrix} \nabla_{u_k} \tilde{g}_k(w_k^0, u_k^0, v_k) & I \end{bmatrix} \end{aligned} \quad (2.250b)$$

$$\begin{aligned} b_k &= g_k(x_k^0, u_k^0) - x_{k+1}^0 = \begin{bmatrix} \tilde{g}_k(w_k^0, u_k^0, v_k) - w_{k+1}^0 \\ u_k^0 - u_k^0 \end{bmatrix} \\ &= \begin{bmatrix} \tilde{g}_k(w_k^0, u_k^0, v_k) - w_{k+1}^0 \\ 0 \end{bmatrix} \end{aligned} \quad (2.250c)$$

Linearization of the inequality constraint

$$h_k(x_k, u_k) \geq d_k \quad (2.251)$$

with

$$h_k(x_k, u_k) = \tilde{h}_k(\Phi' x_k, u_k, \Gamma' x_k + u_k) = \tilde{h}_k(w_k, u_k, \Delta u_k) \quad (2.252)$$

may be expressed as

$$h_k(x_k^0, u_k^0) + \nabla_{x_k} h_k(x_k^0, u_k^0)' \hat{x}_k + \nabla_{u_k} h_k(x_k^0, u_k^0)' \hat{u}_k \geq d_k \quad (2.253)$$

This relation may also be expressed as

$$C_k' \hat{x}_k + D_k' \hat{u}_k + c_k \geq d_k \quad (2.254)$$

in which

$$\begin{aligned} C_k &= \nabla_{x_k} h_k(x_k^0, u_k^0) = \nabla_{x_k} \tilde{h}_k(\Phi' x_k^0, u_k^0, \Gamma' x_k^0 + u_k^0) \\ &= \Phi \nabla_{w_k} \tilde{h}_k(w_k^0, u_k^0, \Delta u_k^0) + \Gamma \nabla_{\Delta u_k} \tilde{h}_k(w_k^0, u_k^0, \Delta u_k^0) \\ &= \begin{bmatrix} \nabla_{w_k} \tilde{h}_k(w_k^0, u_k^0, \Delta u_k^0) \\ -\nabla_{\Delta u_k} \tilde{h}_k(w_k^0, u_k^0, \Delta u_k^0) \end{bmatrix} \end{aligned} \quad (2.255a)$$

$$\begin{aligned} D_k &= \nabla_{u_k} h_k(x_k^0, u_k^0) = \nabla_{u_k} \tilde{h}_k(\Phi' x_k^0, u_k^0, \Gamma' x_k^0 + u_k^0) \\ &= \nabla_{u_k} \tilde{h}_k(w_k^0, u_k^0, \Delta u_k^0) + \nabla_{\Delta u_k} \tilde{h}_k(w_k^0, u_k^0, \Delta u_k^0) \end{aligned} \quad (2.255b)$$

$$c_k = h_k(x_k^0, u_k^0) = \tilde{h}_k(\Phi' x_k^0, u_k^0, \Gamma' x_k^0 + u_k^0) = \tilde{h}_k(w_k^0, u_k^0, \Delta u_k^0) \quad (2.255c)$$

The inequality constraint

$$H_N(x_N) \geq \bar{d}_N \quad (2.256)$$

may be linearized by first order Taylor approximation

$$H_N(x_N^0) + \nabla_{x_N} H_N(x_N^0)' \hat{x}_N \geq \bar{d}_N \quad (2.257)$$

which may also be expressed as

$$C'_N \hat{x}_N + c_N \geq d_N \quad (2.258)$$

in which

$$C_N = \nabla_{x_N} H_N(x_N^0) \quad (2.259a)$$

$$c_N = H_N(x_N^0) \quad (2.259b)$$

$$d_N = \bar{d}_N \quad (2.259c)$$

The objective function of the nonlinear least-squares moving horizon control problem is

$$\phi = \sum_{k=0}^{N-1} l_k(x_k, u_k) + L_N(x_N) \quad (2.260)$$

in which the stage costs,  $l_k(x_k, u_k)$ , are of the least-squares type

$$l_k(x_k, u_k) = \frac{1}{2} \|\psi_k(x_k, u_k)\|_{\Lambda_k}^2 = \frac{1}{2} \psi_k(x_k, u_k)' \Lambda_k \psi_k(x_k, u_k) \quad (2.261)$$

with the residual function,  $\psi_k(x_k, u_k)$ , given by

$$\begin{aligned} \psi_k(x_k, u_k) &= \begin{bmatrix} \tilde{\varphi}_k(w_k, u_k) - z_k \\ \Delta u_k \end{bmatrix} \\ &= \begin{bmatrix} \tilde{\varphi}_k(w_k, u_k) \\ u_k - u_{k-1} \end{bmatrix} = \begin{bmatrix} \tilde{\varphi}_k(\Phi' x_k, u_k) - z_k \\ \Gamma' x_k + u_k \end{bmatrix} \end{aligned} \quad (2.262)$$

and the weight matrix  $\Lambda_k$  given by

$$\Lambda_k = \begin{bmatrix} \tilde{\Lambda}_k & 0 \\ 0 & \tilde{S}_k \end{bmatrix} \quad (2.263)$$

Appendix A describes the construction of the quadratic program used for generating the search direction in a sequential quadratic programming algorithm. Under the Gauss-Newton assumption, the quadratic function generated from the nonlinear least-squares optimal control problem will have the structure

$$\phi = \sum_{k=0}^{N-1} l_k(\hat{x}_k, \hat{u}_k) + l_N(\hat{x}_N) \quad (2.264)$$

in which

$$\begin{aligned} \hat{l}_k(\hat{x}_k, \hat{u}_k) &= \frac{1}{2} \begin{bmatrix} \hat{x}_k \\ \hat{u}_k \end{bmatrix}' \begin{bmatrix} Q_k & M_k \\ M'_k & R_k \end{bmatrix} \begin{bmatrix} \hat{x}_k \\ \hat{u}_k \end{bmatrix} + \begin{bmatrix} q_k \\ r_k \end{bmatrix}' \begin{bmatrix} \hat{x}_k \\ \hat{u}_k \end{bmatrix} + f_k \\ &= \frac{1}{2} (\hat{x}_k' Q_k \hat{x}_k + 2 \hat{x}_k' M_k \hat{u}_k + \hat{u}_k' R_k \hat{u}_k) + q_k' \hat{x}_k + r_k' \hat{u}_k + f_k \end{aligned} \quad (2.265)$$

and

$$l_N(\hat{x}_N) = \frac{1}{2} \hat{x}_N' P_N \hat{x}_N + p_N' \hat{x}_N + \gamma_N \quad (2.266)$$

Let

$$y_k^0 = \tilde{\varphi}_k(w_k^0, u_k^0) \quad (2.267)$$

Then the zero-order term in  $l_k(x_k, u_k)$  around the current iterate  $\{x_{k+1}^0, u_k^0\}_{k=0}^{N-1}$  is

$$f_k = l_k(x_k^0, u_k^0) = \frac{1}{2}(y_k^0 - z_k)' \tilde{\Lambda}_k (y_k^0 - z_k) + \frac{1}{2} \Delta u_k^0 \tilde{S}_k \Delta u_k^0 \quad (2.268)$$

The first order terms are obtained as the gradients of the objective function in the nonlinear least-squares optimal control problem. Due to the separability of the objective function in optimal control problems and the least-squares structure of the stage costs, the parameters associated with the first order terms in  $l_k(\hat{x}_k, \hat{u}_k)$  are

$$\begin{aligned} q_k &= \nabla_{x_k} l_k(x_k^0, u_k^0) = \nabla_{x_k} \psi_k(x_k^0, u_k^0) \Lambda_k \psi_k(x_k^0, u_k^0) \\ &= \begin{bmatrix} \nabla_{w_k} \tilde{\varphi}_k(w_k^0, u_k^0) & 0 \\ 0 & -I \end{bmatrix} \begin{bmatrix} \tilde{\Lambda}_k & 0 \\ 0 & \tilde{S}_k \end{bmatrix} \begin{bmatrix} y_k^0 - z_k \\ \Delta u_k^0 \end{bmatrix} \\ &= \begin{bmatrix} \nabla_{w_k} \tilde{\varphi}_k(w_k^0, u_k^0) \tilde{\Lambda}_k (y_k^0 - z_k) \\ -\tilde{S}_k \Delta u_k^0 \end{bmatrix} \end{aligned} \quad (2.269a)$$

$$\begin{aligned} r_k &= \nabla_{u_k} l_k(x_k^0, u_k^0) = \nabla_{u_k} \psi_k(x_k^0, u_k^0) \Lambda_k \psi_k(x_k^0, u_k^0) \\ &= [\nabla_{u_k} \tilde{\varphi}_k(w_k^0, u_k^0) \quad I] \begin{bmatrix} \tilde{\Lambda}_k & 0 \\ 0 & \tilde{S}_k \end{bmatrix} \begin{bmatrix} y_k^0 - z_k \\ \Delta u_k^0 \end{bmatrix} \\ &= \nabla_{u_k} \tilde{\varphi}_k(w_k^0, u_k^0) \tilde{\Lambda}_k (y_k^0 - z_k) + \tilde{S}_k \Delta u_k^0 \end{aligned} \quad (2.269b)$$

Under the Gauss-Newton assumption, the parameters associated with the quadratic terms of  $l(\hat{x}_k, \hat{u}_k)$  may be computed by (see appendix A)

$$\begin{aligned} Q_k &= \nabla_{x_k} \psi_k(x_k^0, u_k^0) \Lambda_k \nabla_{x_k} \psi_k(x_k^0, u_k^0)' \\ &= \begin{bmatrix} \nabla_{w_k} \tilde{\varphi}_k(w_k^0, u_k^0) & 0 \\ 0 & -I \end{bmatrix} \begin{bmatrix} \tilde{\Lambda}_k & 0 \\ 0 & \tilde{S}_k \end{bmatrix} \begin{bmatrix} \nabla_{w_k} \tilde{\varphi}_k(w_k^0, u_k^0) & 0 \\ 0 & -I \end{bmatrix}' \\ &= \begin{bmatrix} \nabla_{w_k} \tilde{\varphi}_k(w_k^0, u_k^0) \tilde{\Lambda}_k \nabla_{w_k} \tilde{\varphi}_k(w_k^0, u_k^0)' & 0 \\ 0 & \tilde{S}_k \end{bmatrix} \end{aligned} \quad (2.270a)$$

$$\begin{aligned} M_k &= \nabla_{x_k} \psi_k(x_k^0, u_k^0) \Lambda_k \nabla_{u_k} \psi_k(x_k^0, u_k^0)' \\ &= \begin{bmatrix} \nabla_{w_k} \tilde{\varphi}_k(w_k^0, u_k^0) & 0 \\ 0 & -I \end{bmatrix} \begin{bmatrix} \tilde{\Lambda}_k & 0 \\ 0 & \tilde{S}_k \end{bmatrix} [\nabla_{u_k} \tilde{\varphi}_k(w_k^0, u_k^0) \quad I]' \\ &= \begin{bmatrix} \nabla_{w_k} \tilde{\varphi}_k(w_k^0, u_k^0) \tilde{\Lambda}_k \nabla_{u_k} \tilde{\varphi}_k(w_k^0, u_k^0)' \\ -\tilde{S}_k \end{bmatrix} \end{aligned} \quad (2.270b)$$

$$\begin{aligned} R_k &= \nabla_{u_k} \psi_k(x_k^0, u_k^0) \Lambda_k \nabla_{u_k} \psi_k(x_k^0, u_k^0)' \\ &= [\nabla_{u_k} \tilde{\varphi}_k(w_k^0, u_k^0) \quad I] \begin{bmatrix} \tilde{\Lambda}_k & 0 \\ 0 & \tilde{S}_k \end{bmatrix} [\nabla_{u_k} \tilde{\varphi}_k(w_k^0, u_k^0) \quad I]' \\ &= \nabla_{u_k} \tilde{\varphi}_k(w_k^0, u_k^0) \tilde{\Lambda}_k \nabla_{u_k} \tilde{\varphi}_k(w_k^0, u_k^0)' + \tilde{S}_k \end{aligned} \quad (2.270c)$$



The cost,  $L_N(x_N)$ , associated with the final state in the nonlinear least-squares moving horizon control problem is

$$\begin{aligned} L_N(x_N) &= \frac{1}{2} \|x_N - x_s\|_P^2 = \frac{1}{2} (x_N - x_s)' P (x_N - x_s) \\ &= \frac{1}{2} x_N' P x_N - x_s' P x_N + \frac{1}{2} x_s' P x_s \end{aligned} \quad (2.271)$$

As this is a quadratic function itself, it is trivial to recognize that under the Gauss-Newton assumption, the parameters in

$$l_N(\hat{x}_N) = \frac{1}{2} \hat{x}_N' P_N \hat{x}_N + p_N' \hat{x}_N + \gamma_N \quad (2.272)$$

are

$$P_N = P \quad p_N = -P x_s \quad \gamma_N = \frac{1}{2} x_s' P x_s \quad (2.273)$$

Consequently, it has been demonstrated that the quadratic program obtained in an SQP algorithm for the nonlinear least-squares optimal control problem is a constrained linear-quadratic optimal control problem

$$\min_{\{\hat{x}_{k+1}, \hat{u}_k\}_{k=0}^{N-1}} \phi = \sum_{k=0}^{N-1} l_k(\hat{x}_k, \hat{u}_k) + l_N(\hat{x}_N) \quad (2.274a)$$

$$s.t. \quad \hat{x}_{k+1} = A_k' \hat{x}_k + B_k' \hat{u}_k + b_k \quad k = 0, 1, \dots, N-1 \quad (2.274b)$$

$$C_k' \hat{x}_k + D_k' \hat{u}_k + c_k \geq d_k \quad k = 0, 1, \dots, N-1 \quad (2.274c)$$

$$C_N' \hat{x}_N + c_N \geq d_N \quad (2.274d)$$

in which the parameters are obtained by the equations derived in this proof and stated in the proposition.  $\square$

By the discussions and propositions in this section, it has been demonstrated that the major computational operation in computing the approximate control action of the infinite-horizon optimal control problem concerns solution of a constrained linear-quadratic optimal control problem. The data matrices in the constrained linear-quadratic optimal control problem may under the Gauss-Newton assumption be computed according to proposition 2.6.3.

### 2.6.3 Moving Horizon Control with Linear Models

An important special case of the nonlinear least-squares moving horizon controller (2.212) is the linear moving horizon controller. The distinction of the linear moving horizon controller compared to the nonlinear least-squares moving horizon controller is that the constraints in the linear moving horizon controller are affine functions. One formulation of the linear moving horizon controller is

$$\min_{\{w_{k+1}, v_k, y_k\}} \phi = \sum_{k=0}^{\infty} l_k(y_k, \Delta v_k) \quad (2.275a)$$

$$s.t. \quad w_{k+1} = \tilde{A}'_k w_k + \tilde{B}'_k v_k + \tilde{b}_k \quad (2.275b)$$

$$y_k = \tilde{C}'_k w_k + \tilde{c}_k \quad (2.275c)$$

$$v_{min} \leq v_k \leq v_{max} \quad (2.275d)$$

$$-\Delta_v \leq \Delta v_k \leq \Delta_v \quad (2.275e)$$

$$\tilde{d}_k \leq \tilde{G}'_k w_k + \tilde{H}'_k v_k + \tilde{h}_k \leq \tilde{e}_k \quad (2.275f)$$

in which the stage costs are

$$\tilde{l}_k(y_k, \Delta v_k) = \frac{1}{2}(y_k - z_k)' \tilde{\Lambda}_k (y_k - z_k) + \frac{1}{2} \Delta v_k' \tilde{S}_k \Delta v_k \quad (2.276)$$

#### Assumption 2.6.4

The linear moving horizon controller (2.275) has the following finite parameterization

$$\begin{aligned} \mathcal{C} &= \{\mathcal{M}_k\}_{k=0}^{\infty} \\ &= \{\mathcal{M}_0, \mathcal{M}_1, \dots, \mathcal{M}_{N-1}, \mathcal{M}, \mathcal{M}, \dots\} \end{aligned} \quad (2.277a)$$

$$\{z_k\}_{k=0}^{\infty} = \{z_0, z_1, \dots, z_{N-1}, z, z, \dots\} \quad (2.277b)$$

in which

$$\mathcal{M}_k = \{\tilde{A}_k, \tilde{B}_k, \tilde{b}_k, \tilde{C}_k, \tilde{c}_k, \mathcal{V}, \tilde{H}_k, \tilde{G}_k, \tilde{h}_k, \tilde{e}_k, \tilde{d}_k\} \quad (2.278a)$$

$$\mathcal{M} = \{\tilde{A}, \tilde{B}, \tilde{b}, \tilde{C}, \tilde{c}, \mathcal{V}, \tilde{H}, \tilde{G}, \tilde{h}, \tilde{e}, \tilde{d}\} \quad (2.278b)$$

and  $\mathcal{V} = \{v_{min}, v_{max}, \Delta_v\}$ .

#### Assumption 2.6.5

A feasible point  $\{w_{k+1}, v_k\}$  of (2.275) exists.

#### Remark 2.6.6

Scokaert and Rawlings (1999) and Rao and Rawlings (1999) discuss situations in which (2.275) is not feasible and relax (2.275) by introducing soft and exact soft constraints. However, these relaxations do change the structure of the problem.

An approximate solution of (2.275) may be obtained numerically by decomposing the problem into a target problem and a finite horizon regulation problem. The target problem of the linear moving horizon controller (2.275) is

$$\min_{w_s, v_s, y_s} \phi_s = \frac{1}{2}(y_s - z)' \tilde{\Lambda}(y_s - z) + \frac{1}{2}(v_s - \bar{v}_s)' R_s (v_s - \bar{v}_s) \quad (2.279a)$$

$$s.t. \quad w = \tilde{A}' w + \tilde{B}' v + \tilde{b} \quad (2.279b)$$

$$y = \tilde{C}' w + \tilde{c} \quad (2.279c)$$

$$v_{min} \leq v \leq v_{max} \quad (2.279d)$$

$$\tilde{d} \leq \tilde{G}' w + \tilde{H}' v + \tilde{h} \leq \tilde{e} \quad (2.279e)$$

Muske (1997) provides a procedure for computation of the matrix  $R_s$  such that priority is given to meeting the target  $z$  and the auxiliary target  $\bar{v}_s$  does not degrade the performance with respect to the primary target  $z$ . This procedure consists of the following sequence of computations

$$\tilde{N} = \begin{bmatrix} \tilde{N}_w \\ \tilde{N}_v \end{bmatrix} = \text{Null}([I - \tilde{A}' \quad -\tilde{B}']) \quad (2.280a)$$

$$\alpha = \text{Null}(\tilde{N}_w' \tilde{C} \tilde{C}' \tilde{N}_w) \quad (2.280b)$$

$$R_s = \tilde{R} \tilde{N}_v \alpha \alpha' \tilde{N}_v \tilde{R} \quad (2.280c)$$

$\tilde{R}$  is desired weight matrix for the case of no primary target  $z$  and  $\bar{v}_s$  is the target for the inputs.  $\bar{v}_s$  may be selected as the previous input, i.e.  $\bar{v}_s = v_{-1}$ .

The linear least-squares moving horizon controller (2.275) may be transformed to an infinite-horizon linear-quadratic optimal control problem with upper and lower constraints. This transformation is described in proposition 2.6.7. Corollary 2.6.8 shows that the infinite-horizon linear-quadratic optimal control problem with upper and lower constraints is equivalent with an infinite-horizon constrained linear-quadratic optimal control problem.

**Proposition 2.6.7**

Let

$$x_k = \begin{bmatrix} w_k \\ v_{k-1} \end{bmatrix} \quad u_k = v_k \quad (2.281)$$

Then the linear moving horizon controller (2.275) may be expressed as the following infinite-horizon linear quadratic optimal control problem

$$\min_{\{x_{k+1}, u_k\}} \phi = \sum_{k=0}^{\infty} l_k(x_k, u_k) \quad (2.282a)$$

$$s.t. \quad x_{k+1} = A'_k x_k + B'_k u_k + b_k \quad (2.282b)$$

$$\bar{d}_k \leq \bar{C}'_k x_k + \bar{D}'_k u_k + \bar{c}_k \leq \bar{e}_k \quad (2.282c)$$

$$u_{min} \leq u_k \leq u_{max} \quad (2.282d)$$

in which the stage costs are

$$l_k(x_k, u_k) = \frac{1}{2} (x'_k Q_k x_k + 2x'_k M_k u_k + u'_k R_k u_k) + q'_k x_k + r'_k u_k + f_k \quad (2.283)$$

The parameters of the stage costs are

$$Q_k = \begin{bmatrix} \tilde{C}_k \tilde{\Lambda}_k \tilde{C}'_k & 0 \\ 0 & \tilde{S}_k \end{bmatrix} \quad M_k = \begin{bmatrix} 0 \\ -\tilde{S}_k \end{bmatrix} \quad R_k = \tilde{S}_k \quad (2.284a)$$

$$q_k = \begin{bmatrix} \tilde{C}_k \tilde{\Lambda}_k (\bar{c}_k - z_k) \\ 0 \end{bmatrix} \quad r_k = 0 \quad (2.284b)$$

$$f_k = \frac{1}{2} (\bar{c}_k - z_k)' \tilde{\Lambda}_k (\bar{c}_k - z_k) \quad (2.284c)$$

The parameters of the state transition equation are

$$A'_k = \begin{bmatrix} \tilde{A}'_k & 0 \\ 0 & 0 \end{bmatrix} \quad B'_k = \begin{bmatrix} \tilde{B}'_k \\ I \end{bmatrix} \quad b_k = \begin{bmatrix} \tilde{b}_k \\ 0 \end{bmatrix} \quad (2.285)$$

The parameters of the inequality constraints are

$$\bar{C}'_k = \begin{bmatrix} \tilde{G}'_k & 0 \\ 0 & -I \end{bmatrix} \quad \bar{D}'_k = \begin{bmatrix} \tilde{H}'_k \\ I \end{bmatrix} \quad \bar{c}_k = \begin{bmatrix} \tilde{h}_k \\ 0 \end{bmatrix} \quad (2.286a)$$

and

$$\bar{d}_k = \begin{bmatrix} \tilde{d}_k \\ -\Delta_v \end{bmatrix} \quad \bar{e}_k = \begin{bmatrix} \tilde{e}_k \\ \Delta_v \end{bmatrix} \quad (2.286b)$$

$$u_{min} = v_{min} \quad u_{max} = v_{max} \quad (2.286c)$$

The initial state  $x_0$  is a parameter of (2.282) and is

$$x_0 = \begin{bmatrix} w_0 \\ v_{-1} \end{bmatrix} \quad (2.287)$$

*Proof.* Define the augmented state  $x_k$  by

$$x_k = \begin{bmatrix} w_k \\ v_{k-1} \end{bmatrix} \quad (2.288)$$

and define  $u_k = v_k$ . Then

$$w_k = \begin{bmatrix} I \\ 0 \end{bmatrix}' \begin{bmatrix} w_k \\ v_{k-1} \end{bmatrix} = \Phi' x_k \quad (2.289a)$$

$$\Delta v_k = v_k - v_{k-1} = \begin{bmatrix} 0 \\ -I \end{bmatrix}' \begin{bmatrix} w_k \\ v_{k-1} \end{bmatrix} + v_k = \Gamma' x_k + u_k \quad (2.289b)$$

in which

$$\Phi = \begin{bmatrix} I \\ 0 \end{bmatrix} \quad \Gamma = \begin{bmatrix} 0 \\ -I \end{bmatrix}$$

The relation for  $y_k$  may be rearranged to

$$y_k = \tilde{C}'_k w_k + \tilde{c}_k = \tilde{C}'_k \Phi' x_k + \tilde{c}_k = C'_k x_k + c_k \quad (2.290)$$

in which

$$C_k = \Phi \tilde{C}_k = \begin{bmatrix} I \\ 0 \end{bmatrix} \tilde{C}_k = \begin{bmatrix} \tilde{C}_k \\ 0 \end{bmatrix} \quad (2.291a)$$

$$c_k = \tilde{c}_k \quad (2.291b)$$

By the above expressions for  $y_k$  and  $\Delta v_k$ , the stage costs (2.276) of the least-squares moving horizon control problem with linear models (2.275) may be

rearranged as

$$\begin{aligned}
l_k(y_k, \Delta v_k) &= \frac{1}{2}(y_k - z_k)' \tilde{\Lambda}_k (y_k - z_k) + \frac{1}{2} \Delta v_k' \tilde{S}_k \Delta v_k \\
&= \frac{1}{2}(C_k' x_k + c_k - z_k)' \tilde{\Lambda}_k (C_k' x_k + c_k - z_k) \\
&\quad + \frac{1}{2}(\Gamma' x_k + u_k)' \tilde{S}_k (\Gamma' x_k + u_k) \\
&= \frac{1}{2} x_k' \left( C_k \tilde{\Lambda}_k C_k' + \Gamma \tilde{S}_k \Gamma' \right) x_k + x_k' \left( \Gamma \tilde{S}_k \right) u_k + \frac{1}{2} u_k' \tilde{S}_k u_k \quad (2.292) \\
&\quad + (c_k - z_k)' \tilde{\Lambda}_k C_k' x_k + \frac{1}{2} (c_k - z_k)' \tilde{\Lambda}_k (c_k - z_k) \\
&= \frac{1}{2} (x_k' Q_k x_k + 2x_k' M_k u_k + u_k' R_k u_k) + q_k' x_k + r_k' u_k + f_k \\
&= l_k(x_k, u_k)
\end{aligned}$$

in which the parameters associated with the quadratic terms are

$$\begin{aligned}
Q_k &= C_k \tilde{\Lambda}_k C_k' + \Gamma \tilde{S}_k \Gamma' = \begin{bmatrix} \tilde{C}_k \\ 0 \end{bmatrix} \tilde{\Lambda}_k \begin{bmatrix} \tilde{C}_k \\ 0 \end{bmatrix}' + \begin{bmatrix} 0 \\ -I \end{bmatrix} \tilde{S}_k \begin{bmatrix} 0 \\ -I \end{bmatrix}' \\
&= \begin{bmatrix} \tilde{C}_k \tilde{\Lambda}_k \tilde{C}_k' & 0 \\ 0 & \tilde{S}_k \end{bmatrix} \quad (2.293a)
\end{aligned}$$

$$M_k = \Gamma \tilde{S}_k = \begin{bmatrix} 0 \\ -I \end{bmatrix} \tilde{S}_k = \begin{bmatrix} 0 \\ -\tilde{S}_k \end{bmatrix} \quad (2.293b)$$

$$R_k = \tilde{S}_k \quad (2.293c)$$

and the parameters associated with the linear terms are

$$q_k = C_k \tilde{\Lambda}_k (c_k - z_k) = \begin{bmatrix} \tilde{C}_k \\ 0 \end{bmatrix} \tilde{\Lambda}_k (\tilde{c}_k - z_k) = \begin{bmatrix} \tilde{C}_k \tilde{\Lambda}_k (\tilde{c}_k - z_k) \\ 0 \end{bmatrix} \quad (2.293d)$$

$$r_k = 0 \quad (2.293e)$$

The parameter representing the zero order term is

$$f_k = \frac{1}{2} (\tilde{c}_k - z_k)' \tilde{\Lambda}_k (\tilde{c}_k - z_k) \quad (2.293f)$$

Using the state transition equation of (2.275)

$$w_{k+1} = \tilde{A}_k' w_k + \tilde{B}_k' v_k + \tilde{b}_k \quad (2.294)$$

we obtain the state transition equation for the augmented system

$$\begin{aligned}
x_{k+1} &= \begin{bmatrix} w_{k+1} \\ v_k \end{bmatrix} = \begin{bmatrix} \tilde{A}_k' w_k + \tilde{B}_k' v_k + \tilde{b}_k \\ v_k \end{bmatrix} \\
&= \begin{bmatrix} \tilde{A}_k' & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} w_k \\ v_{k-1} \end{bmatrix} + \begin{bmatrix} \tilde{B}_k' \\ I \end{bmatrix} v_k + \begin{bmatrix} \tilde{b}_k \\ 0 \end{bmatrix} \\
&= A_k' x_k + B_k' u_k + b_k \quad (2.295)
\end{aligned}$$

in which we have introduced the parameters

$$A'_k = \begin{bmatrix} \tilde{A}'_k & 0 \\ 0 & 0 \end{bmatrix} \quad B'_k = \begin{bmatrix} \tilde{B}'_k \\ I \end{bmatrix} \quad b_k = \begin{bmatrix} \tilde{b}_k \\ 0 \end{bmatrix}$$

The definition,  $u_k = v_k$ , implies that the constraints

$$v_{\min} \leq v_k \leq v_{\max} \quad (2.296)$$

may be expressed as

$$u_{\min} \leq u_k \leq u_{\max} \quad (2.297)$$

in which

$$u_{\min} = v_{\min} \quad (2.298a)$$

$$u_{\max} = v_{\max} \quad (2.298b)$$

The constraints

$$-\Delta_v \leq \Delta v_k \leq \Delta_v \quad (2.299a)$$

$$\tilde{d}_k \leq \tilde{G}'_k w_k + \tilde{H}'_k v_k + \tilde{h}_k \leq \tilde{e}_k \quad (2.299b)$$

may be expressed as

$$\tilde{d}_k \leq \begin{bmatrix} \tilde{G}'_k & 0 \end{bmatrix} x_k + \tilde{H}'_k u_k + \tilde{h}_k \leq \tilde{e}_k \quad (2.300a)$$

$$-\Delta_v \leq \Gamma' x_k + u_k \leq \Delta_v \quad (2.300b)$$

which is equivalent to

$$\begin{bmatrix} \tilde{d}_k \\ -\Delta_v \end{bmatrix} \leq \begin{bmatrix} \tilde{G}'_k & 0 \\ 0 & -I \end{bmatrix} x_k + \begin{bmatrix} \tilde{H}'_k \\ I \end{bmatrix} u_k + \begin{bmatrix} \tilde{h}_k \\ 0 \end{bmatrix} \leq \begin{bmatrix} \tilde{e}_k \\ \Delta_v \end{bmatrix} \quad (2.301)$$

These inequality relations may be expressed in the compact notation

$$\bar{d}_k \leq \bar{C}'_k x_k + \bar{D}'_k u_k + \bar{c}_k \leq \bar{e}_k \quad (2.302)$$

by definition of the parameters

$$\bar{C}'_k = \begin{bmatrix} \tilde{G}'_k & 0 \\ 0 & -I \end{bmatrix} \quad \bar{D}'_k = \begin{bmatrix} \tilde{H}'_k \\ I \end{bmatrix} \quad \bar{c}_k = \begin{bmatrix} \tilde{h}_k \\ 0 \end{bmatrix} \quad (2.303)$$

as well as

$$\bar{d}_k = \begin{bmatrix} \tilde{d}_k \\ -\Delta_v \end{bmatrix} \quad \bar{e}_k = \begin{bmatrix} \tilde{e}_k \\ \Delta_v \end{bmatrix} \quad (2.304)$$

By the above deduction, it is clear that (2.275) can be expressed as (2.282) with the parameters computed as stated in the proposition.  $\square$

**Corollary 2.6.8**

The infinite-horizon linear quadratic optimal control problem (2.282) is equivalent with the following infinite-horizon linear quadratic optimal control problem

$$\min_{\{x_{k+1}, u_k\}} \quad \phi = \sum_{k=0}^{\infty} l_k(x_k, u_k) \quad (2.305a)$$

$$s.t. \quad x_{k+1} = A'_k x_k + B'_k u_k + b_k \quad (2.305b)$$

$$C'_k x_k + D'_k u_k + c_k \geq d_k \quad (2.305c)$$

in which

$$C'_k = \begin{bmatrix} \bar{C}'_k \\ 0 \\ -\bar{C}'_k \\ 0 \end{bmatrix} \quad D'_k = \begin{bmatrix} \bar{D}'_k \\ I \\ -\bar{D}'_k \\ -I \end{bmatrix} \quad c_k = \begin{bmatrix} \tilde{c}_k \\ 0 \\ -\tilde{c}_k \\ 0 \end{bmatrix} \quad d_k = \begin{bmatrix} \bar{d}_k \\ u_{min} \\ -\bar{e}_k \\ u_{max} \end{bmatrix} \quad (2.306)$$

*Proof.* Follows directly by simple algebraic manipulations turning all inequality constraints into lower bounded inequality constraints.  $\square$

An approximate solution of the infinite-horizon linear moving horizon controller (2.275) is obtained by computing an approximate solution to the equivalent infinite-horizon constrained linear quadratic optimal control problem (2.305). The approximation is constructed by selecting a sufficiently long finite horizon  $N$  and solving the finite horizon constrained linear quadratic optimal control problem

$$\min_{\{x_{k+1}, u_k\}_{k=0}^{N-1}} \quad \phi = \sum_{k=0}^{N-1} l_k(x_k, u_k) + L_N(x_N) \quad (2.307a)$$

$$s.t. \quad x_{k+1} = A'_k x_k + B'_k u_k + b_k, \quad k \in \mathcal{N} \quad (2.307b)$$

$$C'_k x_k + D'_k u_k + c_k \geq d_k, \quad k \in \mathcal{N} \quad (2.307c)$$

$$H'_N x_N + h_N \geq g_N \quad (2.307d)$$

in which the final stage cost is

$$\begin{aligned} L_N(x_N) &= \frac{1}{2} (x_N - x_s)' P (x_N - x_s) \\ &= \frac{1}{2} x_N' P_N x_N + p_N' x_N + \gamma_N \end{aligned} \quad (2.308)$$

$x_s = [w'_s, v'_s]'$  is the optimal steady-state computed as the solution of the target problem (2.279). By this choice of  $L_N(x_N)$  the parameters  $P_N$ ,  $p_N$ , and  $\gamma_N$  becomes

$$P_N = P \quad p_N = -P x_s \quad \gamma_N = \frac{1}{2} x_s' P x_s \quad (2.309)$$

Hence, it has been established that computing a solution to the linear moving horizon controller involves solving a constrained linear quadratic optimal control problem (2.30). Once again, the demand for efficient numerical solution procedures for the constrained linear-quadratic optimal control problem has been illustrated.

Chmielewski and Manousiouthakis (1996) as well as Scokaert and Rawlings (1998) address the construction of the finite-horizon approximation (2.307) of the infinite-horizon constrained linear quadratic control problem (2.305). Under the assumption that the target  $(x_s, y_s) = (w_s, v_s, y_s)$  is in the interior of the maximal output admissible set (c.f. Gilbert and Tan, 1991; Blanchini, 1999), i.e. no inequality constraints are active at steady state, they prove that a finite horizon  $N$  exists such that (2.307) is an exact approximation of (2.305) when the matrix  $P$  is selected as the solution of the Riccati equation

$$P = Q + APA' - (M + APB')(R + BPB')^{-1}(M + APB')' \quad (2.310)$$

in which the matrices for the particular model considered (2.275) are

$$Q = \begin{bmatrix} \tilde{C}\tilde{\Lambda}\tilde{C}' & 0 \\ 0 & \tilde{S} \end{bmatrix} \quad M = \begin{bmatrix} 0 \\ -\tilde{S} \end{bmatrix} \quad R = \tilde{S} \quad (2.311a)$$

$$A = \begin{bmatrix} \tilde{A} & 0 \\ 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} \tilde{B} & I \end{bmatrix} \quad (2.311b)$$

For a more restricted class of problems than considered here, Chmielewski and Manousiouthakis (1996) computes an  $N$  guaranteeing that the finite horizon approximation is exact. Scokaert and Rawlings (1998) solves a sequence of finite-horizon constrained linear quadratic control problems, i.e. (2.307) without the final constraint (2.307d). The horizon  $N$  is sufficient for an exact approximation, when the final optimal state  $x_N^*$  belongs to the maximal output admissible set.

Motivated by the cases presented by Rao and Rawlings (1999), Pannocchia *et al.* (2002) relax the assumption that  $(x_s, u_s) = (w_s, v_s, y_s)$  must be in the interior and allows input constraints to be active at steady state. In this approach, they solve two sequences of constrained linear quadratic optimal control problems (2.307) with different choices of  $P$ . These sequences provide a converging sequence of upper and lower bounds on the value of the infinite-horizon problem. These sequences also bound the numerical error on the computed finite-horizon optimal solution compared to the infinite-horizon optimal solution. Hence, by solving sequences of constrained linear quadratic optimal control problems a solution to the infinite horizon problem is obtained for any specified accuracy.

## 2.6.4 Anticipatory Control

Anticipatory control is an extension and generalization of feed forward control. In feedback control, the controller is provided with the current state as estimated from the available measurements and a set-point which tacitly is assumed to be constant. In addition a feedforward controller is given information about some current exogeneous process inputs. An anticipatory



controller is provided information about the expected future sequence of set-points  $\{z_k\}_{k=0}^{\infty}$  as well as the expected future sequence of exogeneous process inputs  $\{\zeta_k\}_{k=0}^{\infty}$ . Model predictive control is ideally suited for implementation of anticipatory control and has been proposed in a number of studies (c.f. De Keyser *et al.*, 1988; Eaton and Rawlings, 1992; Mosca and Casavola, 1995; Zhu *et al.*, 2001; Bemporad *et al.*, 2002). Provided, sufficiently accurate forecasts,  $\{z_k\}_{k=0}^{\infty}$  and  $\{\zeta_k\}_{k=0}^{\infty}$ , are available, anticipatory model predictive control is an efficient methodology to suspend the performance limitations of feedback control applied to processes with right half plane zeros and time-delays and achieve nominally perfect control (c.f. Eaton and Rawlings, 1992; Morari, 1983; Jørgensen and Jørgensen, 2000).

In a moving horizon implementation of anticipatory control, the anticipated sequences,  $\{z_k\}_{k=0}^{\infty}$  and  $\{\zeta_k\}_{k=0}^{\infty}$ , are updated at each sample time and are conveniently specified by the finite parameterizations

$$\{z_k\}_{k=0}^{\infty} = \{z_0, z_1, \dots, z_{N-1}, z, z, \dots\} \quad (2.312a)$$

$$\{\zeta_k\}_{k=0}^{\infty} = \{\zeta_0, \zeta_1, \dots, \zeta_{N-1}, \zeta, \zeta, \dots\} \quad (2.312b)$$

The set point sequence  $\{z_k\}_{k=0}^{\infty}$  enter the moving horizon controller through the stage costs of the objective function. In the general nonlinear moving horizon controller (2.197) the stage costs are specified as  $\tilde{f}_k(w_k, u_k, \Delta u_k, z_k)$ . Similarly, in the least-squares nonlinear moving horizon controller (2.212) the stage costs are specified as  $\tilde{l}_k(y_k, \Delta u_k, z_k)$  and as  $\tilde{l}_k(y_k, \Delta v_k, z_k)$  for the linear moving horizon controller. Hence, the sequence  $\{z_k\}_{k=0}^{\infty}$  enters naturally in the moving horizon controller. As is evident by (2.240d), (2.240e) and (2.284b),  $\{z_k\}_{k=0}^{\infty}$  affects the coefficients  $q_k$  and  $r_k$  of the linear terms in the objective function of the constrained linear quadratic control problem (2.30) that is solved in the respective cases. Consequently, extension of the stage costs with linear terms, i.e.  $q'_k x_k + r'_k u_k$ , facilitates anticipatory moving horizon control.

The anticipated exogeneous process input sequence,  $\{\zeta_k\}_{k=0}^{\infty}$ , enters the nonlinear moving horizon controllers, (2.197) and (2.212), through the state transition equation  $w_{k+1} = \tilde{g}_k(w_k, u_k, \zeta_k)$ . Hence, in nonlinear model predictive control, the anticipated sequence  $\{\zeta_k\}_{k=0}^{\infty}$  enters simply in the function specifying the state transition and does not complicate the controller much. The principle structure of the state transition equation in a linear moving horizon controller with known exogeneous process inputs,  $\{\zeta_k\}_{k=0}^{\infty}$ , is

$$\begin{aligned} x_{k+1} &= A'_k x_k + B'_k u_k + \tilde{b}_k + E'_k \zeta_k \\ &= A'_k x_k + B'_k u_k + b_k \end{aligned} \quad (2.313)$$

in which  $b_k = \tilde{b}_k + E'_k \zeta_k$ . Consequently, the specification of the linear anticipative moving horizon controller is facilitated by having an affine term,  $b_k$ , in the state transition equation.

Anticipative model predictive control with its ability to incorporate expected exogeneous process inputs  $\{\zeta_k\}_{k=0}^{\infty}$  enables coordinated decentralized model

predictive control (c.f. Zhu *et al.*, 2000; Zhu and Henson, 2002). Coordinated decentralized model predictive control decomposes a plant model into submodels each controlled by model predictive control. The coordination consists of informing each decentral model predictive controller about the expected process consequences of the actions of the other controllers. The coordination procedure is a current research topic, but it is expected to be well suited for decentralized model predictive control of integrated processing systems (see also Garcia and Morari, 1981, 1984).

## 2.7 Conclusion

The moving horizon methodology embedded in model predictive control for solution of a stochastic optimal control problem has been discussed and the inherent assumptions pinpointed. The explicit statement and consideration of these assumptions may be the starting point for development of numerical tractable controllers accounting more accurately for the stochastics of process systems than currently applied model predictive controllers. Moving horizon estimation and control constitute a suboptimal solution to the stochastic optimal control problem. To obtain numerical tractability, moving horizon estimation and control separates the stochastic optimal control problem into an estimation problem and a deterministic optimal control problem. This decomposition is based on an ad hoc application of the separation theorem and the certainty-equivalence principle which are valid for linear-quadratic Gaussian systems. However, for general systems the separation theorem and the certainty-equivalence principle constitute approximations which make the resulting control system suboptimal.

It has been demonstrated that both the nonlinear moving horizon estimation problem as well as the nonlinear moving horizon control problem are instances of a deterministic constrained optimal control problem. In each iteration of the sequential quadratic programming algorithm for solution of the deterministic constrained optimal control problem, a constrained linear-quadratic optimal control problem must be solved. Structured modified BFGS updates may be used to retain the linear-quadratic optimal control structure of the quadratic-program generated in each iteration of quasi-Newton SQP algorithms. For least-squares problems the very efficient generation of the data for the constrained linear-quadratic optimal control problem under the Gauss-Newton assumption has been discussed. Also in the linear case, the numerical solution of both the moving horizon estimator and controller at each sample time must be obtained by solving a constrained linear-quadratic optimal control problem. In contrast to standard practice, it is important to retain the affine term in the state transition equation as well as the linear terms in the stage costs of the linear-quadratic optimal control problem. These developments motivates tailored algorithms for solution of the constrained linear-quadratic optimal con-

trol problem. The application of Riccati recursions to solve the linear-quadratic optimal control problem efficiently has been indicated.

Finally, it has been indicated how the affine term in the state transition equation and the linear terms in the stage costs of the linear-quadratic optimal control problem facilitates anticipatory control and coordinated decentralized model predictive control.

The constrained linear-quadratic optimal control problem occupies a central enabling role in numerical realization of model predictive control for large-scale nonlinear systems.

# Numerical Methods for Moving Horizon Estimation and Control

The linear moving horizon estimator and controller may both be solved by solving a linear quadratic optimal control problem. A primal active set, a dual active set, and an interior point algorithm for solution of the linear quadratic optimal control problem are presented. The major computational effort in all these algorithms reduces to solution of certain unconstrained linear quadratic optimal control problems. A Riccati recursion procedure for effective solution of such unconstrained problems is stated.

## 3.1 Introduction

Model predictive control has been established as the preferred advanced control technology in the process industries. Its success is attributed to its ability to handle hard constraints and its ability to use plant models identified by standard techniques. The implementation of model predictive control *systems* requires repeated on-line solution of a state estimation problem and an optimal control problem. Both the state estimator and the optimal controller can be formulated as constrained optimization problems (Allgöwer *et al.*, 1999; Binder *et al.*, 2001a). Formulated as optimization problems and implemented in a moving horizon manner, the estimator is called a moving horizon estimator (MHE) and the controller is called a model predictive controller (MPC) or a moving horizon controller. Due to the requirement of real-time solution of both the estimation and control problem, the range and scale of processes that can be controlled by model predictive control systems depends critically on the ability to solve the constrained optimization problems efficiently and reliably.

In this paper we outline numerical methods for efficient solution of constrained moving horizon estimation and control problems for systems described by affine models and with quadratic objective functions. Both the moving horizon esti-

mator and controller of such systems are formulated as sparse convex quadratic programs with equality and inequality constraints. In contrast, it is possible to eliminate the state variables by the dynamic model and obtain dense convex quadratic programs with inequalities only. Both interior-point and active set algorithms for the solution of the quadratic program corresponding to a linear-quadratic optimal control based on a banded-matrix factorization has been proposed for solution of the model predictive control problem (Wright, 1996). The active set algorithm suggested in Wright (1996) is of the primal type and the linear algebra is based on updating an LU-factorization each time a constraint is added to or removed from the current working set of active constraints. The quadratic programs corresponding to the moving horizon controller and estimator, respectively, have been solved by interior-point algorithms using Riccati iterations for the matrix factorization (Rao *et al.*, 1998; Tenny and Rawlings, 2002). The Riccati factorization is very efficient for solution of the KKT system as it exploits the specific block-diagonal structure arising in the moving horizon control and estimation problem. Alternatively, a dual active set algorithm using the Schur-complement and a general sparse matrix factorization has been suggested for solution of the model predictive control problem (Bartlett *et al.*, 2000).

We show that in the primal active-set, the dual active-set, and the interior-point algorithm for general convex quadratic programming problems, the search direction may be computed by solving a Karush-Kuhn-Tucker (KKT) system with the same structure as an equality constrained quadratic program. In quadratic programs of constrained optimal control problems, the KKT-system corresponding to the unconstrained control problem has a special structure and may be solved efficiently by a Riccati recursion. Both the constrained moving horizon estimation problem and the moving horizon control problem may be formulated as an optimal control problem and solved efficiently using the Riccati recursion.

The key contribution of this paper is a dual algorithm based on Riccati based factorization for solution of the quadratic program constituting a constrained linear-quadratic optimal control problem. It is demonstrated how the constrained linear-quadratic optimal control problem may represent a moving horizon controller and a moving horizon estimator. Furthermore, the importance and key role of the KKT-system in numerical solution of the constrained-linear quadratic optimal control problem by the primal active set algorithm as well as the dual active set algorithm and the interior-point algorithm is emphasized and explained.

## 3.2 Algorithms for Solution of Convex QPs

Consider the general convex quadratic program

$$\min_{x \in \mathbb{R}^n} f(x) = \frac{1}{2}x'Gx + g'x \quad (3.1a)$$

$$s.t. \quad A'x = b \quad (3.1b)$$

$$C'x \geq d \quad (3.1c)$$

in which the Hessian matrix  $G \in \mathbb{R}^{n \times n}$  is symmetric and positive semi-definite. Assume further that the matrix  $A \in \mathbb{R}^{n \times m}$  has full column rank and that the KKT-matrix

$$\begin{bmatrix} G & -A \\ -A' & 0 \end{bmatrix} \quad (3.2)$$

is non-singular. Further let the columns of the matrix  $C \in \mathbb{R}^{n \times m_I}$  be denoted  $c_i \in \mathbb{R}^n$  for  $i \in \mathcal{I} = \{1, 2, \dots, m_I\}$ . The Lagrangian of (3.1) is

$$\mathcal{L}(x, \pi, \lambda) = \frac{1}{2}x'Gx + g'x - \pi'(A'x - b) - \lambda'(C'x - d) \quad (3.3)$$

and the necessary and sufficient conditions for optimality of (3.1) are

$$Gx + g - A\pi - \sum_{i \in \mathcal{A}(x)} c_i \lambda_i = 0 \quad (3.4a)$$

$$A'x = b \quad (3.4b)$$

$$C'x \geq d \quad (3.4c)$$

$$\lambda_i \geq 0 \quad i \in \mathcal{A}(x) = \{i \in \mathcal{I} : c'_i x = d_i\} \quad (3.4d)$$

$$\lambda_i = 0 \quad i \in \mathcal{I} \setminus \mathcal{A}(x) \quad (3.4e)$$

Active set algorithms for solution of (3.1) apply the conditions (3.4) in the search for the minimizer of (3.1). A current working set  $\mathcal{W} \subset \mathcal{A}(x)$  of active constraints is recurrent and the search direction is constructed by utilizing  $\lambda_i = 0$  for  $i \in \mathcal{I} \setminus \mathcal{W}$  and  $c'_i x = d_i$  for  $i \in \mathcal{W}$  and partly ignoring the remaining inequalities.

At a primal feasible point, primal active set algorithms compute the primal search direction  $p$  along which the objective function decreases (Gill and Murray, 1978). Along with the search direction  $p$ , the equality constraint Lagrange multipliers  $\pi$ , and the Lagrange multipliers  $\lambda$  associated with the current working set of active constraints are computed.  $(p, \pi, \lambda)$  are computed as the solution of

$$\begin{bmatrix} G & -A & -F \\ -A' & 0 & 0 \\ -F' & 0 & 0 \end{bmatrix} \begin{bmatrix} p \\ \pi \\ \lambda \end{bmatrix} = - \begin{bmatrix} Gx + g \\ 0 \\ 0 \end{bmatrix} \quad F = [c_i]_{i \in \mathcal{W}} \quad (3.5)$$

The next iterate is  $\bar{x} = x + \alpha p$  in which the step length  $\alpha$  is selected to maintain primal feasibility, i.e.  $c'_i \bar{x} \geq d_i$  for all  $i \in \mathcal{I} \setminus \mathcal{W}$ . Optimality of the current

iterate is determined by the sign of  $\lambda$ . The current iterate is optimal if  $\lambda \geq 0$  and  $p = 0$ . At each iteration either an inequality constraint is appended to or removed from the current working set of active constraints. This implies that a column is appended to the matrix  $F$  when an inequality constraint is appended to the current working set of active constraints. A column is removed from the matrix  $F$  when a constraint is removed from the current working set. Accordingly, the KKT-matrix at each iteration changes by a single column and row. Efficient active set algorithms exploit this simple modification of the KKT-matrix such that it is not refactorized at each iteration, but rather its factorization is updated. Initially, a primal feasible point must be provided as primal active set algorithms proceed from primal feasible points and maintains primal feasibility while improving the value of the program. Determination of the feasible point, for instance by solving a phase-I linear program, may be a substantial part of the overall computations in determining an optimal point.

Primal active set algorithms proceed by solving a sequence of equality constrained quadratic programs in which some of the inequality constraints are treated as equalities. Simultaneously, these algorithms generate the iterates such that primal feasibility is maintained. Dual active set algorithms proceed by generating iterates such that the objective function of the dual program is improved while maintaining dual feasibility (Goldfarb and Idnani, 1983). The dual program of (3.1) is

$$\max_{x, \pi, \lambda} \mathcal{L}(x, \pi, \lambda) \quad (3.6a)$$

$$s.t. \quad Gx - A\pi - C\lambda = -g \quad (3.6b)$$

$$\lambda \geq 0 \quad (3.6c)$$

Let inequality constraint  $r$ , i.e.  $c'_r x \geq d_r$ , be violated. In the dual active set algorithm, the search direction  $(p, w, v)$  along which the dual objective function increases is computed as the solution to the KKT system

$$\begin{bmatrix} G & -A & -F \\ -A' & 0 & 0 \\ -F' & 0 & 0 \end{bmatrix} \begin{bmatrix} p \\ w \\ v \end{bmatrix} = \begin{bmatrix} c_r \\ 0 \\ 0 \end{bmatrix} \quad F = [c_i]_{i \in \mathcal{W}} \quad (3.7)$$

The next iterate  $(\bar{x}, \bar{\pi}, \bar{\lambda})$  is computed by  $\bar{x} = x + \alpha p$ ,  $\bar{\pi} = \pi + \alpha w$ ,  $\bar{\lambda}_{\mathcal{W}} = \lambda_{\mathcal{W}} + \alpha v$ ,  $\bar{\lambda}_r = \lambda_r + \alpha$ .  $\lambda_{\mathcal{W}}$  is the Lagrange multipliers associated to the inequality constraints in the current working set of primal active constraints (and dual inactive constraints). The step length  $\alpha$  is selected such that dual feasibility, i.e.  $\lambda \geq 0$ , is maintained. By construction of the search direction, the constraint (3.6b) is satisfied as  $\lambda_i = 0$  for  $i \in \mathcal{I} \setminus \mathcal{W}$ . A particular advantage of the dual active set algorithm, is that an initial dual feasible point is readily available. It may be obtained as the solution to KKT-system of the equality constrained primal program (3.1a)-(3.1b), i.e. as the solution of

$$\begin{bmatrix} G & -A \\ -A' & 0 \end{bmatrix} \begin{bmatrix} x \\ \pi \end{bmatrix} = - \begin{bmatrix} g \\ b \end{bmatrix} \quad (3.8)$$

Interior-point algorithms for solution of (3.1) are not based on the optimality conditions (3.4) but on necessary and sufficient optimality conditions in the following form

$$Gx + g - A\pi - C\lambda = 0 \quad (3.9a)$$

$$A'x = b \quad (3.9b)$$

$$C'x \geq d \quad (3.9c)$$

$$\lambda \geq 0 \quad (3.9d)$$

$$\lambda_i(c'_i x - d_i) = 0 \quad i \in \mathcal{I} \quad (3.9e)$$

Instead of the active set condition,  $\lambda_i \geq 0$  for  $i \in \mathcal{A}(x)$  and  $\lambda_i = 0$  for  $i \in \mathcal{I} \setminus \mathcal{A}(x)$ , the formulation (3.9) is based on the complementarity condition (3.9e). Introduce slack variables,  $t = C'x - d \geq 0$ , and the notation

$$T = \begin{bmatrix} t_1 & & \\ & \ddots & \\ & & t_{m_I} \end{bmatrix} \quad \Lambda = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_{m_I} \end{bmatrix} \quad e = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \quad (3.10)$$

such that the conditions (3.9) may be expressed as

$$Gx + g - A\pi - C\lambda = 0 \quad (3.11a)$$

$$A'x - b = 0 \quad (3.11b)$$

$$C'x - d - t = 0 \quad (3.11c)$$

$$T\Lambda e = 0 \quad (3.11d)$$

$$(t, \lambda) \geq 0 \quad (3.11e)$$

These conditions may be regarded as a system of nonlinear equations represented as  $F(x, \pi, \lambda, t) = 0$  with the requirement  $(t, \lambda) \geq 0$ . The Mehrotra predictor-corrector algorithm is an interior-point method which computes the search direction as a combination of a predictor and a corrector step (Rao *et al.*, 1998). At both the predictor and corrector step  $t$  and  $\lambda$  are maintained in the interior, i.e.  $(t, \lambda) > 0$ . The predictor step is a pure Newton step for  $F(x, \pi, \lambda, t)$ , i.e. (3.11a)-(3.11d), while the corrector step is a modified Newton step. In both cases, the structure of the equations solved in computing the search direction is

$$\begin{bmatrix} G & -A & -C & 0 \\ -A' & 0 & 0 & 0 \\ -C' & 0 & 0 & I \\ 0 & 0 & T_k & \Lambda_k \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta \pi \\ \Delta \lambda \\ \Delta t \end{bmatrix} = - \begin{bmatrix} r_G \\ r_A \\ r_C \\ r_\Lambda \end{bmatrix} \quad (3.12)$$

### 3.3 KKT-Systems

The major computation in algorithms for general convex quadratic programs (3.1) is computation of the search direction. This corresponds to solution of



(3.5) in the primal active set algorithm, (3.7) in the dual active set algorithm, and (3.12) in the interior point algorithm, respectively.

Consider, the equality constrained quadratic subproblem of (3.1)

$$\min_{x \in \mathbb{R}^n} f(x) = \frac{1}{2}x'Gx + g'x \quad (3.13a)$$

$$s.t. \quad A'x = b \quad (3.13b)$$

The solution of this program is computed as the solution of the following KKT system

$$\begin{bmatrix} G & -A \\ -A' & 0 \end{bmatrix} \begin{bmatrix} x \\ \pi \end{bmatrix} = - \begin{bmatrix} g \\ b \end{bmatrix} \quad (3.14)$$

The solution of (3.5), (3.7), and (3.12) may essentially be reduced to solution of systems with the structure (3.14). This is advantageous if (3.14) has a special structure that facilitates its efficient computation.

**Proposition 3.3.1 (Schur-complement solution of KKT-system)**

Let  $G \in \mathbb{R}^{n \times n}$  be symmetric positive semi-definite. Let  $\begin{bmatrix} A & F \end{bmatrix} \in \mathbb{R}^{n \times (m+m_F)}$  have full column rank. Let the KKT-matrix (3.2) be non-singular. Then

$$S = \begin{bmatrix} F' & 0 \end{bmatrix} \begin{bmatrix} G & -A \\ -A' & 0 \end{bmatrix}^{-1} \begin{bmatrix} F \\ 0 \end{bmatrix} \quad (3.15)$$

is symmetric positive definite and has the Cholesky factorization  $S = LL'$ . Furthermore, the unique solution  $(p, s, u)$  of

$$\begin{bmatrix} G & -A & -F \\ -A' & 0 & 0 \\ -F' & 0 & 0 \end{bmatrix} \begin{bmatrix} p \\ s \\ u \end{bmatrix} = - \begin{bmatrix} h \\ 0 \\ 0 \end{bmatrix} \quad (3.16)$$

may be obtained by solving the following sequence of equations

$$\begin{bmatrix} G & -A \\ -A' & 0 \end{bmatrix} \begin{bmatrix} p_0 \\ s_0 \end{bmatrix} = - \begin{bmatrix} h \\ 0 \end{bmatrix} \quad (3.17a)$$

$$LL'u = -F'p_0 \quad (3.17b)$$

$$\begin{bmatrix} G & -A \\ -A' & 0 \end{bmatrix} \begin{bmatrix} \Delta p \\ \Delta s \end{bmatrix} = - \begin{bmatrix} -Fu \\ 0 \end{bmatrix} \quad (3.17c)$$

$$\begin{bmatrix} p \\ s \end{bmatrix} = \begin{bmatrix} p_0 \\ s_0 \end{bmatrix} + \begin{bmatrix} \Delta p \\ \Delta s \end{bmatrix} \quad (3.17d)$$

*Proof.* See Ouellette (1981). □

Proposition 3.3.1 may be used for solution of (3.5) in the primal active set algorithm and solution of (3.7) in the dual active set algorithm. In the first case  $h = Gx + g$  and in the latter case  $h = -c_r$ . The matrix  $S$  is not computed from scratch at each iteration. Rather, its Cholesky factorization is updated utilizing that the matrix  $F$  changes by a single column at each iteration in the active

set algorithms. The Cholesky factor,  $L$ , is treated as a dense matrix. Hence, the method is most efficient when only a few inequality constraints are active at the optimal solution. The method is only efficient, when the KKT-matrix (3.2) has a sparse structure that can be utilized. The KKT matrix (3.2) used in proposition 3.3.1 remains constant and need only to be factorized once.

**Proposition 3.3.2 (Interior-Point KKT System)**

Let  $G \in \mathbb{R}^{n \times n}$  be symmetric positive semi-definite. Let (3.2) be non-singular. Let  $(t_k, \lambda_k) > 0$ . Then the unique solution  $(\Delta x, \Delta \pi, \Delta \lambda, \Delta t)$  of (3.12) may be obtained by computation of the unique solution of

$$\begin{bmatrix} G + CT_k^{-1}\Lambda_k C' & -A \\ -A' & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta \pi \end{bmatrix} = - \begin{bmatrix} \tilde{r}_G \\ r_A \end{bmatrix} \quad (3.18a)$$

in which

$$\tilde{r}_G = -r_G + CT_k^{-1}(-r_\Lambda + \Lambda_k r_C) \quad (3.18b)$$

and subsequent computation of

$$\Delta t = -r_C + C' \Delta x \quad (3.18c)$$

$$\Delta \lambda = T_k^{-1}(-r_\Lambda - \Lambda_k \Delta t) \quad (3.18d)$$

*Proof.* Follows by simple rearrangement of (3.12). See Rao *et al.* (1998).  $\square$

As is evident from proposition 3.3.2, the computational burden in finding the search direction of the interior-point algorithm is solution of (3.18a). The structure of (3.18a) is identical to the structure of (3.14), and therefore the computation of the search direction in the interior-point algorithm is facilitated by efficient solution of (3.14). The KKT-matrix in (3.18a) changes as  $(\lambda_k, t_k)$  changes and must therefore be refactorized at each iteration.

## 3.4 Linear Quadratic Optimal Control

Proposition 3.3.1 and 3.3.2 provide methodologies for solution of inequality and equality constrained convex quadratic programming. The efficiency of these methods depends on the efficiency of the solution method for the corresponding equality constrained convex quadratic program.

The unconstrained linear quadratic optimal control problem is the equality constrained convex quadratic program

$$\min_{\{x_{k+1}, u_k\}_{k=0}^{N-1}} \phi = \sum_{k=0}^{N-1} l_k(x_k, u_k) + l_N(x_N) \quad (3.19a)$$

$$s.t. \quad x_{k+1} = A'_k x_k + B'_k u_k + b_k \quad (3.19b)$$

in which the stage costs of the objective function are

$$l_k(x_k, u_k) = \frac{1}{2} (x_k' Q_k x_k + 2x_k' M_k u_k + u_k' R_k u_k) + q_k' x_k + r_k' u_k + f_k \quad (3.20a)$$

$$l_N(x_N) = \frac{1}{2} x_N' P_N x_N + p_N' x_N + \gamma_N \quad (3.20b)$$

The matrices  $\begin{bmatrix} Q_k & M_k \\ M_k' & R_k \end{bmatrix}$  and  $P_N$  are assumed to be symmetric positive semi-definite and the KKT-matrix of the problem is assumed to be non-singular.

The necessary and sufficient optimality conditions for (3.19) are

$$Q_k x_k + M_k u_k + q_k - \pi_{k-1} + A_k \pi_k = 0 \quad (3.21a)$$

$$M_k' x_k + R_k u_k + r_k + B_k \pi_k = 0 \quad (3.21b)$$

$$P_N x_N + p_N - \pi_{N-1} = 0 \quad (3.21c)$$

$$x_{k+1} = A_k' x_k + B_k' u_k + b_k \quad (3.21d)$$

In the case  $N = 2$ , the necessary and sufficient optimality conditions may be expressed as the KKT-system

$$\left[ \begin{array}{ccc|cc} R_0 & & & B_0 & \\ & Q_1 & M_1 & -I & A_1 \\ & M_1' & R_1 & & B_1 \\ & & & P_2 & -I \\ \hline B_0' & -I & & & \\ & A_1' & B_1' & -I & \end{array} \right] \begin{bmatrix} u_0 \\ x_1 \\ u_1 \\ x_2 \\ \pi_0 \\ \pi_1 \end{bmatrix} = - \begin{bmatrix} r_0 + M_0' x_0 \\ q_1 \\ r_1 \\ p_2 \\ b_0 + A_0' x_0 \\ b_1 \end{bmatrix}$$

which may be rearranged to

$$\left[ \begin{array}{cccccc} R_0 & B_0 & & & & \\ B_0' & 0 & -I & & & \\ & -I & Q_1 & M_1 & A_1 & \\ & & M_1' & R_1 & B_1 & \\ & & A_1' & B_1' & 0 & -I \\ & & & & -I & P_2 \end{array} \right] \begin{bmatrix} u_0 \\ \pi_0 \\ x_1 \\ u_1 \\ \pi_1 \\ x_2 \end{bmatrix} = - \begin{bmatrix} r_0 + M_0' x_0 \\ b_0 + A_0' x_0 \\ q_1 \\ r_1 \\ b_1 \\ p_2 \end{bmatrix}$$

The necessary and sufficient conditions (5.87) for optimality of (3.19) are sparse and highly structured. The following proposition prescribes a Riccati iteration procedure for solution of (5.87).

**Proposition 3.4.1 (Linear Quadratic Optimal Control Solution)**

Let  $x_0$ ,  $\{A_k, B_k, b_k, Q_k, M_k, R_k, q_k, r_k\}_{k=0}^{N-1}$ , and  $\{P_N, p_N\}$  be given. Then the solution  $\{u_k, \pi_k, x_{k+1}\}_{k=0}^{N-1}$  of (5.87) may be obtained by the following procedure

1. Compute

$$K_k = -(R_k + B_k P_{k+1} B'_k)^{-1} (M'_k + B_k P_{k+1} A'_k) \quad (3.22a)$$

$$a_k = -(R_k + B_k P_{k+1} B'_k)^{-1} (r_k + B_k (P_{k+1} b_k + p_{k+1})) \quad (3.22b)$$

$$P_k = Q_k + A_k P_{k+1} A'_k + (M_k + A_k P_{k+1} B'_k) K_k \quad (3.22c)$$

$$p_k = (A_k + K'_k B_k) (P_{k+1} b_k + p_{k+1}) + q_k + K'_k r_k \quad (3.22d)$$

for  $k = N-1, N-2, \dots, 0$ .

2. Compute the primal solution  $\{u_k, x_{k+1}\}_{k=0}^{N-1}$  for  $k = 0, 1, \dots, N-1$  by

$$u_k = K_k x_k + a_k \quad (3.23a)$$

$$x_{k+1} = A'_k x_k + B'_k u_k + b_k \quad (3.23b)$$

3. Obtain the dual solution  $\{\pi_k\}_{k=0}^{N-1}$  by computing

$$\pi_{N-1} = P_N x_N + p_N \quad (3.24a)$$

$$\pi_{k-1} = A_k \pi_k + Q_k x_k + M_k u_k + q_k \quad (3.24b)$$

for  $k = N-1, N-2, \dots, 1$ .

*Proof.* See Rao *et al.* (1998) or Ravn (1999).  $\square$

Let  $x \in \mathbb{R}^n$  and  $u \in \mathbb{R}^m$ . Then this method has complexity  $O(N(n^3 + m^3))$  while a dense method on the same KKT-system has complexity  $O(N^3(n+m)^3)$ . Corresponding dense quadratic programs obtained by elimination of the states has complexity  $O(N^3 m^3)$ . The Riccati based factorization is thus two order of magnitudes faster than the dense based approach when the horizon  $N$  length is much larger than the state dimension  $n$ .

The constrained linear quadratic optimal control problem is (3.19) with the additional constraints

$$C'_k x_k + D'_k u_k + c_k \geq d_k \quad k = 0, 1, \dots, N-1 \quad (3.25a)$$

$$C'_N x_N + c_N \geq d_N \quad (3.25b)$$

The search direction in algorithms for solution of this problem is computed efficiently by combination of proposition 3.4.1 and either proposition 3.3.1 or 3.3.2.

The model predictive controller and the moving horizon estimator are particular instances of the optimal control problem, i.e. (3.19) and (3.25). For simplicity and to focus on the essentials, this is demonstrated for the model predictive controller and moving horizon estimator without inequality constraints. These formulations are easily extended to the inequality constrained cases.

### 3.5 Model Predictive Control

The unconstrained model predictive controller for linear systems may be expressed as

$$\min_{\{y_k, w_{k+1}, u_k\}_{k=0}^{\infty}} \quad \phi = \sum_{k=0}^{\infty} \tilde{l}_k(y_k, \Delta u_k) \quad (3.26a)$$

$$s.t. \quad w_{k+1} = \tilde{A}'_k w_k + \tilde{B}'_k u_k + \tilde{b}_k \quad (3.26b)$$

$$y_k = \tilde{C}'_k w_k + \tilde{c}_k \quad (3.26c)$$

in which the stage cost is

$$\tilde{l}_k(y_k, \Delta u_k) = \frac{1}{2}(y_k - z_k)' \tilde{Q}_k (y_k - z_k) + \frac{1}{2} \Delta u_k \tilde{S}_k \Delta u_k \quad (3.27)$$

The goal stated by this cost, is to keep the systems output  $\{y_k\}_{k=0}^{\infty}$  close to some prescribed trajectory  $\{z_k\}_{k=0}^{\infty}$  while simultaneously limiting the actuator variation  $\Delta u_k = u_k - u_{k-1}$ . Let the data at stage  $k$  of this controller be  $\mathcal{M}_k = \{\tilde{A}_k, \tilde{B}_k, \tilde{b}_k, \tilde{C}_k, \tilde{c}_k, \tilde{Q}_k, \tilde{S}_k\}$  and let  $\mathcal{M} = \{\tilde{A}, \tilde{B}, \tilde{b}, \tilde{C}, \tilde{c}, \tilde{Q}, \tilde{S}\}$ . Assume that the controller (3.26) is parameterized as

$$\mathcal{C} = \{\mathcal{M}_k\}_{k=0}^{\infty} = \{\mathcal{M}_0, \mathcal{M}_1, \dots, \mathcal{M}_{N-1}, \mathcal{M}, \mathcal{M}, \dots\} \quad (3.28)$$

and that the reference trajectory has the parameterization

$$\{z_k\}_{k=0}^{\infty} = \{z_0, z_1, \dots, z_{N-1}, z, z, \dots\} \quad (3.29)$$

The optimal steady state consistent with the controller model (3.26) is obtained by solution of the quadratic program

$$\min_{u, w, y} \quad \frac{1}{2}(y - z)' \tilde{Q} (y - z) + (u - u_s)' R_s (u - u_s) \quad (3.30a)$$

$$s.t. \quad w = \tilde{A}' w + \tilde{B}' u + \tilde{b} \quad (3.30b)$$

$$y = \tilde{C}' w + \tilde{c} \quad (3.30c)$$

in which  $u_s$  is a target of the input if there are degrees of freedom in excess and  $R_s$  is computed by the procedure (Muske, 1997)

$$\tilde{N} = \text{Null}([I - \tilde{A}' \quad -\tilde{B}']) = \begin{bmatrix} \tilde{N}_x \\ \tilde{N}_u \end{bmatrix} \quad (3.31a)$$

$$\alpha = \text{Null}(\tilde{N}_x' \tilde{C} \tilde{C}' \tilde{N}_x) \quad (3.31b)$$

$$R_s = \tilde{R} \tilde{N}_u \alpha \alpha' \tilde{N}_u' \tilde{R} \quad (3.31c)$$

When a target  $(u, w)$  has been computed the model predictive controller algorithm solves the dynamic quadratic program

$$\min \quad \phi = \sum_{k=0}^{N-1} l_k(x_k, u_k) + l_N(x_N) \quad (3.32a)$$

$$s.t. \quad x_{k+1} = A'_k x_k + B'_k u_k + b_k \quad (3.32b)$$

in which

$$x_k = \begin{bmatrix} w_k \\ u_{k-1} \end{bmatrix} \quad A'_k = \begin{bmatrix} \tilde{A}'_k & 0 \\ 0 & 0 \end{bmatrix} \quad B'_k = \begin{bmatrix} \tilde{B}'_k \\ I \end{bmatrix} \quad b_k = \begin{bmatrix} \tilde{b}_k \\ 0 \end{bmatrix} \quad (3.33)$$

and  $l_k(x_k, u_k)$  is of the form (3.20a) with the parameters

$$Q_k = \begin{bmatrix} \tilde{C}_k \tilde{Q}_k \tilde{C}'_k & 0 \\ 0 & \tilde{S}_k \end{bmatrix} \quad M_k = \begin{bmatrix} 0 \\ -\tilde{S}_k \end{bmatrix} \quad R_k = \tilde{S}_k \quad (3.34a)$$

$$q_k = \begin{bmatrix} \tilde{C}_k \tilde{Q}_k (\tilde{c}_k - z_k) \\ 0 \end{bmatrix} \quad r_k = 0 \quad (3.34b)$$

$$f_k = \frac{1}{2} (\tilde{c}_k - z_k)' \tilde{Q}_k (\tilde{c}_k - z_k) \quad (3.34c)$$

Let

$$x = \begin{bmatrix} w \\ u \end{bmatrix} \quad Q = \begin{bmatrix} \tilde{C} \tilde{Q} \tilde{C}' & 0 \\ 0 & \tilde{S} \end{bmatrix} \quad M = \begin{bmatrix} 0 \\ -\tilde{S} \end{bmatrix} \quad R = \tilde{S} \quad (3.35)$$

and compute  $P$  from the Riccati equation

$$P = Q + APA' - (M + APB')(R + BPB')^{-1}(M + APB')' \quad (3.36)$$

Then the selected cost-to-go function in (3.32)

$$l_N(x_N) = \frac{1}{2} (x_N - x)' P (x_N - x) \quad (3.37)$$

is identical to (3.20b) with the parameters

$$P_N = P \quad p_N = -Px \quad \gamma_N = \frac{1}{2} x' P x \quad (3.38)$$

Consequently, the dynamic quadratic program of model predictive control is an instance of a linear quadratic optimal control problem.

### 3.6 Moving Horizon Estimation

The unconstrained moving horizon estimation (MHE) problem is the convex quadratic program

$$\min_{\{x_k, w_k, v_k\}} \quad \phi = \tilde{l}_0(x_0) + \sum_{k=0}^N \tilde{l}_k(w_k, v_k) \quad (3.39a)$$

$$s.t. \quad x_{k+1} = A'_k x_k + B'_k w_k + b_k \quad (3.39b)$$

$$y_k = C'_k x_k + v_k + c_k \quad (3.39c)$$

in which the stage costs are the quadratic functions

$$\tilde{l}_0(x_0) = \frac{1}{2}(x_0 - \tilde{x}_0)' \tilde{P}_0^{-1}(x_0 - \tilde{x}_0) \quad (3.40a)$$

$$\tilde{l}_k(w_k, v_k) = \frac{1}{2} \begin{bmatrix} w_k - \tilde{w}_k \\ v_k - \tilde{v}_k \end{bmatrix}' \begin{bmatrix} \tilde{Q}_k & \tilde{M}_k \\ \tilde{M}'_k & \tilde{R}_k \end{bmatrix}^{-1} \begin{bmatrix} w_k - \tilde{w}_k \\ v_k - \tilde{v}_k \end{bmatrix} \quad (3.40b)$$

In the MHE problem, the measurements  $\{y_k\}_{k=0}^N$  are given, and the states  $\{x_k\}_{k=0}^{N+1}$ , the process noise  $\{w_k\}_{k=0}^N$ , and the measurement noise  $\{v_k\}_{k=0}^N$  must be computed by solving (3.39). This corresponds to selecting the states, the process noise, and the measurement noise such that they in some sense give the best fit to the measurements.

Define  $x_{-1} = \tilde{x}_0$  and  $w_{-1} = x_0 - \tilde{x}_0$ . This implies

$$x_0 = x_{-1} + w_{-1} \quad (3.41)$$

and further that  $\tilde{l}_0 = \tilde{l}_0(x_0)$  may be formulated as

$$\tilde{l}_0 = \frac{1}{2} w_{-1}' \tilde{P}_0^{-1} w_{-1} = l_{-1}(x_{-1}, w_{-1}) \quad (3.42)$$

Using the Schur-complement, the weight matrix in (3.40b) may be expressed as

$$\begin{aligned} \begin{bmatrix} W_k & S'_k \\ S_k & V_k \end{bmatrix} &= \begin{bmatrix} \tilde{Q}_k & \tilde{M}_k \\ \tilde{M}'_k & \tilde{R}_k \end{bmatrix}^{-1} \\ &= \begin{bmatrix} \Delta_{\tilde{R}_k}^{-1} & -\Delta_{\tilde{R}_k}^{-1} \tilde{M}_k \tilde{R}_k^{-1} \\ -\tilde{R}_k^{-1} \tilde{M}'_k \Delta_{\tilde{R}_k}^{-1} & \Delta_{\tilde{Q}_k}^{-1} \end{bmatrix} \end{aligned} \quad (3.43)$$

in which  $\Delta_{\tilde{R}_k}$  and  $\Delta_{\tilde{Q}_k}$  are the Schur complements of  $\tilde{R}_k$  and  $\tilde{Q}_k$ , respectively, i.e.

$$\Delta_{\tilde{R}_k} = \tilde{Q}_k - \tilde{M}_k \tilde{R}_k^{-1} \tilde{M}'_k \quad (3.44a)$$

$$\Delta_{\tilde{Q}_k} = \tilde{R}_k - \tilde{M}'_k \tilde{Q}_k^{-1} \tilde{M}_k \quad (3.44b)$$

Applying (3.43), the stage costs (3.40b) may be expressed as

$$\begin{aligned}
\tilde{l}_k &= \frac{1}{2} \begin{bmatrix} w_k - \tilde{w}_k \\ \tilde{y}_k - C'_k x_k \end{bmatrix}' \begin{bmatrix} W_k & S'_k \\ S_k & V_k \end{bmatrix} \begin{bmatrix} w_k - \tilde{w}_k \\ \tilde{y}_k - C'_k x_k \end{bmatrix} \\
&= \frac{1}{2} (x'_k C_k V_k C'_k x_k - 2x'_k C_k S_k w_k + w'_k W_k w_k) \\
&\quad + (C_k (S_k \tilde{w}_k - V_k \tilde{y}_k))' x_k + (S'_k \tilde{y}_k - W_k \tilde{w}_k)' w_k \\
&\quad + \frac{1}{2} (\tilde{w}'_k W_k \tilde{w}_k - 2\tilde{y}'_k S_k \tilde{w}_k + \tilde{y}_k V_k \tilde{y}_k) \\
&= l_k(x_k, w_k)
\end{aligned} \tag{3.45}$$

in which  $\tilde{y}_k$  is defined as

$$\tilde{y}_k = y_k - c_k - \tilde{v}_k \tag{3.46}$$

The MHE problem (3.39) is equivalent to the convex quadratic program

$$\min_{\{x_{k+1}, w_k\}_{k=-1}^{N-1}} \phi = \sum_{k=-1}^N l_k(x_k, w_k) \tag{3.47a}$$

$$s.t. \quad x_0 = x_{-1} + w_{-1} \tag{3.47b}$$

$$x_{k+1} = A'_k x_k + B'_k w_k + b_k \tag{3.47c}$$

which by inspection is recognized as a special version of the linear quadratic optimal control problem (3.19). Consequently, the moving horizon estimation problem (3.39) is a linear quadratic optimal control problem (3.19) with time range  $k = -1$  to  $N + 1$  instead of  $k = 0$  to  $N$ . Furthermore, at time  $k = -1$  the matrices of the MHE-problem in the optimal control problem are

$$Q_{-1} = 0, \quad M_{-1} = 0, \quad R_{-1} = \tilde{P}_0^{-1}, \quad q_{-1} = 0, \quad r_{-1} = 0 \tag{3.48a}$$

$$A_{-1} = I, \quad B_{-1} = I, \quad b_{-1} = 0 \tag{3.48b}$$

At times  $k = 0, 1, \dots, N$ , the weight matrices of the optimal control formulation of the MHE-problem are

$$Q_k = C_k V_k C'_k \quad M_k = -C_k S_k \quad R_k = W_k \tag{3.49a}$$

$$q_k = C_k (S_k \tilde{w}_k - V_k \tilde{y}_k) \quad r_k = S'_k \tilde{y}_k - W_k \tilde{w}_k \tag{3.49b}$$

in which

$$W_k = (\tilde{Q}_k - \tilde{M}_k \tilde{R}_k^{-1} \tilde{M}'_k)^{-1} \tag{3.50a}$$

$$S_k = -\tilde{R}_k^{-1} \tilde{M}'_k (\tilde{Q}_k - \tilde{M}_k \tilde{R}_k^{-1} \tilde{M}'_k)^{-1} \tag{3.50b}$$

$$V_k = (\tilde{R}_k - \tilde{M}'_k \tilde{Q}_k^{-1} \tilde{M}_k)^{-1} \tag{3.50c}$$

The cost-to-go matrices of the MHE-problem formulated as an optimal control problem are

$$P_{N+1} = 0, \quad p_{N+1} = 0 \tag{3.51}$$



### 3.7 Conclusion

Model predictive control and moving horizon estimation with affine models are instances of the linear quadratic optimal control problem.

The search-direction of active-set algorithms for solution of the constrained linear quadratic optimal control problems, (3.19) and the additional constraints (3.25), may be efficiently computed using the decomposition stated in proposition 3.3.1 and the Riccati iteration stated in proposition 3.4.1. When the model parameterization is fixed a priori, as for instance for linear time invariant systems, the factorization of the unconstrained linear quadratic optimal control problem may be done off-line, and the method increases even further in efficiency.

The search direction of the Mehrotra predictor-corrector interior-point algorithm for solution of the constrained linear quadratic optimal control problem may be efficiently computed by combination of proposition 3.3.2 specialized to (3.19,3.25) and proposition 3.4.1.

# Dynamic Programming for Linear-Quadratic Optimal Control

The deterministic linear-quadratic optimal control problem and the deterministic extended linear-quadratic optimal control problem are introduced and defined. The deterministic linear-quadratic optimal control problem is identical with the linear quadratic regulation (LQR) problem. The extended linear-quadratic optimal control problem is an extension of the linear quadratic regulation problem. In addition to the quadratic terms, the objective function of the extended linear-quadratic optimal control problem contains linear terms and inconsequential zero order terms. Rather than being linear, the dynamics of the extended linear-quadratic optimal control problem is described by an affine relation. The extended linear-quadratic optimal control problem has important applications in the solution of the nonlinear optimal control problem as well as in efficient solution algorithms for the constrained linear-quadratic optimal control problem.

Based on dynamic programming, Riccati recursion procedures for the linear-quadratic optimal control problem as well as the extended linear-quadratic optimal control problem are developed. Compared to alternative solution procedures such as control vector parameterization by elimination of the states, the Riccati based procedure is highly efficient for long prediction horizons. The extended linear-quadratic optimal control problem may also be regarded as an equality constrained quadratic program with special structure. The computation of the optimal solution-Lagrange multiplier pair for a convex equality constrained quadratic program is specialized to the extended linear-quadratic optimal control problem treated as a quadratic program. Efficient solution of the highly structured KKT-system corresponding to the extended linear-quadratic optimal control problem is facilitated by the Riccati recursion developed by dynamic programming.

## 4.1 Dynamic Programming

Dynamic programming is the essential principle applied in developing the solution procedures based on Riccati recursion for the linear-quadratic optimal control problem and the extended linear-quadratic optimal control problem. Therefore, the dynamic programming algorithm for deterministic optimal control problem is introduced. Bertsekas (1995a) provides a comprehensive discussion of dynamic programming and its applications for optimal control.

The basic problem studied is the deterministic optimal control problem.

### Problem 4.1.1 (Deterministic Optimal Control)

Let  $x_k \in \mathbb{R}^n$  for  $k = 0, 1, \dots, n$ . Let  $u_k \in \mathbb{R}^m$  for  $k = 0, 1, \dots, N-1$ . Let  $f_k : \mathbb{R}^n \times \mathbb{R}^m \mapsto \mathbb{R}^n$  for  $k = 0, 1, \dots, N-1$ . Let  $\mathcal{U}_k(x_k) \subset \mathbb{R}^m$  for  $k = 0, 1, \dots, N-1$ . Then the deterministic optimal control problem is the mathematical program

$$\min_{\{x_{k+1}, u_k\}_{k=0}^{N-1}} \phi = \sum_{k=0}^{N-1} g_k(x_k, u_k) + g_N(x_N) \quad (4.1a)$$

$$s.t. \quad x_{k+1} = f_k(x_k, u_k) \quad k = 0, 1, \dots, N-1 \quad (4.1b)$$

$$u_k \in \mathcal{U}_k(x_k) \quad k = 0, 1, \dots, N-1 \quad (4.1c)$$

in which  $x_0$  is a parameter. The optimal solution is  $\{x_{k+1}^*, u_k^*\}_{k=0}^{N-1} = \{x_{k+1}^*(x_0), u_k^*(x_0)\}_{k=0}^{N-1}$  and  $\phi^* = \phi^*(x_0)$ .

The optimal solution of the deterministic optimal control problem (4.1) is specified in terms of an optimal sequence,  $\{u_k^*\}_{k=0}^{N-1}$ . In dynamic programming the optimal solution is typically *not* specified as a specific value but rather as a function.

### Definition 4.1.2 (Optimal Policy)

Let  $x_k \in \mathbb{R}^n$ . Let  $\mu_k^* : \mathbb{R}^n \mapsto \Omega_k$  for  $k = 0, 1, \dots, N-1$  in which  $\Omega_k \subset \mathcal{U}_k \subset \mathbb{R}^m$ . Let

$$\phi = \phi(\{x_k\}_{k=0}^N, \{u_k\}_{k=0}^{N-1}) = \sum_{k=0}^{N-1} g_k(x_k, u_k) + g_N(x_N) \quad (4.2)$$

Then  $\pi^*$  defined as the sequence

$$\pi^* = \{\mu_0^*(x_0), \mu_1^*(x_1), \dots, \mu_{N-1}^*(x_{N-1})\} = \{\mu_k^*(x_k)\}_{k=0}^{N-1} \quad (4.3)$$

is an optimal policy of (4.1) if

$$\phi(\{x_k^*\}_{k=0}^N, \{\mu_k^*(x_k^*)\}_{k=0}^{N-1}) \leq \phi(\{x_{k+1}\}_{k=0}^N, \{u_k\}_{k=0}^{N-1}) \quad (4.4)$$

for all  $x_0^* = x_0$ ,  $\{x_{k+1}^* = f(x_k^*, \mu_k^*(x_k^*))\}_{k=0}^{N-1}$ , and  $\{x_{k+1} = f_k(x_k, u_k), u_k \in \mathcal{U}_k(x_k)\}_{k=0}^{N-1}$ .

### Remark 4.1.3

An optimal policy is a function mapping the states to the inputs such that the objective function of (4.1) is minimized for all possible values of the states.

**Proposition 4.1.4 (Principle of Optimality)**

Let  $\pi^* = \{\mu_0^*(x_0), \mu_1^*(x_1), \dots, \mu_{N-1}^*(x_{N-1})\} = \{\mu_k^*(x_k)\}_{k=0}^{N-1}$  be an optimal policy for (4.1).

Then  $\{\mu_i^*(x_i), \mu_{i+1}^*(x_{i+1}), \dots, \mu_{N-1}^*(x_{N-1})\} = \{\mu_k^*(x_k)\}_{k=i}^{N-1}$  is an optimal policy for

$$\min_{\{x_{k+1}, u_k\}_{k=i}^{N-1}} \sum_{k=i}^{N-1} g_k(x_k, u_k) + g_N(x_N) \quad (4.5a)$$

$$s.t. \quad x_{k+1} = f_k(x_k, u_k) \quad k = i, i+1, \dots, N-1 \quad (4.5b)$$

$$u_k \in \mathcal{U}_k(x_k) \quad k = i, i+1, \dots, N-1 \quad (4.5c)$$

*Proof.* This proposition is due to Bellman and can be established by creating a contradiction. Let  $\{\mu_k^*(x_k)\}_{k=0}^{N-1}$  be an optimal policy for (4.1). Assume that  $\{\kappa_k(x_k)\}_{k=i}^{N-1}$  is an optimal policy for (4.5) and assume that  $\{\kappa_k(x_k) \neq \mu_k^*(x_k)\}_{k=i}^{N-1}$ . Let  $x_i \in \mathbb{R}^n$  be any state and let this state be reached from the initial state  $x_0$  by application of the optimal policy  $\{\mu_k^*(x_k)\}_{k=0}^{i-1}$ , i.e.  $\{x_{k+1} = f_k(x_k, \mu_k^*(x_k))\}_{k=0}^{i-1}$ . Then the cost-to-go from  $x_i$  can be minimized by applying the optimal policy  $\{\kappa_k(x_k) \neq \mu_k^*(x_k)\}_{k=i}^{N-1}$ . This implies that

$$\{\mu_0^*(x_0), \mu_1^*(x_1), \dots, \mu_{i-1}^*(x_{i-1}), \kappa_i(x_i), \kappa_{i+1}(x_{i+1}), \dots, \kappa_{N-1}(x_{N-1})\} \quad (4.6)$$

is an optimal policy of (4.1). However, this contradicts the assumption that  $\{\mu_k^*(x_k)\}_{k=0}^{N-1}$  is an optimal policy. Therefore  $\{\kappa_k(x_k) = \mu_k^*(x_k)\}_{k=i}^{N-1}$  and  $\{\mu_k^*(x_k)\}_{k=i}^{N-1}$  is an optimal policy for (4.5).  $\square$

**Remark 4.1.5**

The above proof is not completely rigorous, but the main principles of a rigorous proof are outlined in the arguments. The main complicating issue in doing a rigorous proof is possible non-uniqueness of an optimal policy.

The dynamic programming algorithm stated in the following proposition is based on the principle of optimality.

**Proposition 4.1.6 (Dynamic Programming)**

For every initial state  $x_0 \in \mathbb{R}^n$ , the optimal cost  $\phi^*(x_0)$  to (4.1) is

$$\phi^*(x_0) = V_0(x_0) \quad (4.7)$$

in which the value function  $V_0(x_0)$  is computed by the recursion

$$V_N(x_N) = g_N(x_N) \quad (4.8a)$$

$$V_k(x_k) = \min_{u_k \in \mathcal{U}_k(x_k)} g_k(x_k, u_k) + V_{k+1}(f_k(x_k, u_k)) \quad k = N-1, N-2, \dots, 1, 0 \quad (4.8b)$$

Furthermore, if  $u_k^* = \mu_k^*(x_k)$  minimizes the right hand side of (4.8b) for each  $x_k \in \mathbb{R}^n$  and  $k$ , then the policy  $\pi^* = \{\mu_0^*(x_0), \mu_1^*(x_1), \dots, \mu_{N-1}^*(x_{N-1})\} = \{\mu_k^*(x_k)\}_{k=0}^{N-1}$  is optimal.

*Proof.* Let  $\pi^k$  be defined by

$$\pi^k = \{\mu_k^*(x_k), \mu_{k+1}^*(x_{k+1}), \dots, \mu_{N-1}^*(x_{N-1})\} \quad (4.9)$$

and let  $\pi^k$  denote an optimal policy of

$$\min_{\{x_{i+1}, u_i\}_{i=k}^{N-1}} \phi_k(x_k) = \sum_{i=k}^{N-1} g_i(x_i, u_i) + g_N(x_N) \quad (4.10a)$$

$$s.t. \quad x_{i+1} = f_i(x_i, u_i) \quad i = k, k+1, \dots, N-1 \quad (4.10b)$$

$$u_i \in \mathcal{U}_i(x_i) \quad i = k, k+1, \dots, N-1 \quad (4.10c)$$

The value function,  $V_k(x_k) = \phi_k(x_k)$  of this problem is

$$\begin{aligned} V_k(x_k) &= \min \left\{ \sum_{i=k}^{N-1} g_i(x_i, u_i) + g_N(x_N) : x_{i+1} = f_i(x_i, u_i), u_i \in \mathcal{U}_i(x_i), i = k, k+1, \dots, N-1 \right\} \\ &= \left\{ \sum_{i=k}^{N-1} g_i(x_i, \mu_i^*(x_i)) + g_N(x_N) : x_{i+1} = f_i(x_i, \mu_i^*(x_i)), i = k, k+1, \dots, N-1 \right\} \end{aligned} \quad (4.11)$$

By this definition of  $V_k(x_k)$ , it is clear that  $\phi^*(x_0) = V_0(x_0)$ . Consequently, by definition

$$V_N(x_N) = g_N(x_N) \quad (4.12)$$

and

$$\begin{aligned} V_k(x_k) &= \min \left\{ \sum_{i=k}^{N-1} g_i(x_i, u_i) + g_N(x_N) : x_{i+1} = f_i(x_i, u_i), u_i \in \mathcal{U}_i(x_i), i = k, k+1, \dots, N-1 \right\} \\ &= \min \left\{ g_k(x_k, u_k) + \sum_{i=k+1}^{N-1} g_i(x_i, u_i) + g_N(x_N) : \right. \\ &\quad \left. x_{i+1} = f_i(x_i, u_i), u_i \in \mathcal{U}_i(x_i), i = k, k+1, \dots, N-1 \right\} \\ &= \min \left\{ \min \left\{ \sum_{i=k+1}^{N-1} g_i(x_i, u_i) + g_N(x_N) : x_{i+1} = f_i(x_i, u_i), u_i \in \mathcal{U}_i(x_i), i = k+1, k+2, \dots, N-1 \right\} \right. \\ &\quad \left. + g_k(x_k, u_k) : x_{k+1} = f_k(x_k, u_k), u_k \in \mathcal{U}_k(x_k) \right\} \\ &= \min \{ g_k(x_k, u_k) + V_{k+1}(x_{k+1}) : x_{k+1} = f_k(x_k, u_k), u_k \in \mathcal{U}_k(x_k) \} \\ &= \min_{u_k \in \mathcal{U}_k(x_k)} \{ g_k(x_k, u_k) + V_{k+1}(f_k(x_k, u_k)) \} \end{aligned} \quad (4.13)$$

Furthermore, it is obvious from this derivation that

$$\mu_k^*(x_k) = \arg \min_{u_k \in \mathcal{U}_k(x_k)} g_k(x_k, u_k) + V_k(f_k(x_k, u_k)) \quad (4.14)$$

is an optimal control law.  $\square$

The dynamic programming algorithm is essential in derivation of the Riccati recursion for linear-quadratic optimal control. In fact this application, is one of the most important and successful applications of the dynamic programming algorithm.

## 4.2 Standard Linear-Quadratic Optimal Control

In this section the solution of the time variant deterministic linear quadratic optimal control problem is derived. The solution is given by a Riccati based recursion. The solution and its properties facilitate establishment of corresponding properties of the solution of the extended linear-quadratic optimal control problem.

### Problem 4.2.1 (Linear-Quadratic Optimal Control)

The linear-quadratic optimal control problem consists of solving the quadratic program

$$\min_{\{x_{k+1}, u_k\}_{k=0}^{N-1}} \phi = \sum_{k=0}^{N-1} l_k(x_k, u_k) + l_N(x_N) \quad (4.15a)$$

$$s.t. \quad x_{k+1} = A'_k x_k + B'_k u_k \quad k = 0, 1, \dots, N-1 \quad (4.15b)$$

with the stage costs given by

$$l_k(x_k, u_k) = \frac{1}{2} x'_k Q_k x_k + x'_k M_k u_k + \frac{1}{2} u'_k R_k u_k \quad k = 0, 1, \dots, N-1 \quad (4.16a)$$

$$l_N(x_N) = \frac{1}{2} x'_N P_N x_N \quad (4.16b)$$

In (4.15),  $x_0$  is a parameter and not a decision variable. The optimal solution consists of the minimizer  $(x^*, u^*) = \{x_{k+1}^*, u_k^*\}_{k=0}^{N-1}$  and the optimal value  $\phi^* = \phi(x^*, u^*)$ .

### Remark 4.2.2

The stage costs (4.16a) may also be expressed as

$$\begin{aligned} l_k(x_k, u_k) &= \frac{1}{2} x'_k Q_k x_k + x'_k M_k u_k + \frac{1}{2} u'_k R_k u_k \\ &= \frac{1}{2} \begin{pmatrix} x_k \\ u_k \end{pmatrix}' \begin{pmatrix} Q_k & M_k \\ M'_k & R_k \end{pmatrix} \begin{pmatrix} x_k \\ u_k \end{pmatrix} \quad k = 0, 1, \dots, N-1 \end{aligned} \quad (4.17)$$

### Assumption 4.2.3 (Symmetric Positive Semi-Definiteness)

All matrices

$$\begin{pmatrix} Q_k & M_k \\ M'_k & R_k \end{pmatrix} \quad (4.18)$$

in the sequence

$$\left\{ \begin{pmatrix} Q_k & M_k \\ M'_k & R_k \end{pmatrix} \right\}_{k=0}^{N-1} \quad (4.19)$$

are symmetric positive semi-definite.  $P_N$  is symmetric positive semi-definite.

**Assumption 4.2.4 (Sufficient Uniqueness and Optimality Condition)**

All matrices,  $R_k$ , in the sequence of matrices  $\{R_k\}_{k=0}^{N-1}$  are positive definite.

**Proposition 4.2.5 (Solution of the Linear-Quadratic Optimal Control Problem)**

Consider the linear-quadratic optimal control problem (4.15). Let assumption 4.2.3 and 4.2.4 be satisfied.

The the unique global minimizer,  $\{x_{k+1}^*, u_k^*\}_{k=0}^{N-1}$ , of (4.15) may be obtained by first computing

$$R_{e,k} = R_k + B_k P_{k+1} B_k' \quad (4.20a)$$

$$K_k = -R_{e,k}^{-1}(M_k + A_k P_{k+1} B_k')' \quad (4.20b)$$

$$P_k = Q_k + A_k P_{k+1} A_k' - K_k' R_{e,k} K_k \quad (4.20c)$$

for  $k = N-1, N-2, \dots, 1, 0$  and subsequent computation of

$$u_k^* = K_k x_k^* \quad (4.21a)$$

$$x_{k+1}^* = A_k' x_k^* + B_k' u_k^* \quad (4.21b)$$

for  $k = 0, 1, \dots, N-1$  with  $x_0^* = x_0$ . The optimal value,  $\phi^*$ , of (4.15) is

$$\phi^* = \frac{1}{2} x_0' P_0 x_0 \quad (4.22)$$

*Proof.* Let  $V_k(x_k)$  denote the value function

$$V_k(x_k) = \min_{\{x_{i+1}, u_i\}_{i=k}^{N-1}} \left\{ \sum_{i=k}^{N-1} l_i(x_i, u_i) + l_N(x_N) : x_{i+1} = A_i' x_i + B_i' u_i, i = k, k+1, \dots, N-1 \right\} \quad (4.23)$$

Note that  $V_0(x_0)$  corresponds to the optimal value of (4.15).  $V_k(x_k)$  corresponds to the optimal value of a linear-quadratic optimal control problem starting at stage  $k$  and with the state at stage  $k$ ,  $x_k$ , given. By the above definition of the value function, it is clear that

$$V_N(x_N) = l_N(x_N) = \frac{1}{2} x_N' P_N x_N \quad (4.24)$$

in which  $P_N$  by assumption is a symmetric positive semi-definite matrix. This implies that  $V_N(x_N) \geq 0$  for  $x_N \in \mathbb{R}^n$ . By Bellman's principle of optimality and the dynamic programming algorithm (c.f. Bertsekas, 1995a), it is clear that the value function  $V_k(x_k)$  defined above may be expressed recursively as

$$\begin{aligned} V_k(x_k) &= \min_{x_{k+1}, u_k} \{l_k(x_k, u_k) + V_{k+1}(x_{k+1}) : x_{k+1} = A_k' x_k + B_k' u_k\} \\ &= \min_{u_k} \{l_k(x_k, u_k) + V_{k+1}(A_k' x_k + B_k' u_k)\} \\ &= \min_{u_k} \phi_k(x_k, u_k) \end{aligned} \quad (4.25)$$

in which the function  $\phi_k(x_k, u_k)$  is defined by

$$\phi_k(x_k, u_k) = l_k(x_k, u_k) + V_{k+1}(A'_k x_k + B'_k u_k) \quad (4.26)$$

We want to show that the value function  $V_k(x_k)$  at stage  $k$  is

$$V_k(x_k) = \frac{1}{2} x'_k P_k x_k \quad (4.27)$$

in which  $P_k$  is a symmetric positive semi-definite matrix. This is true for  $k = N$ , i.e.  $V_N(x_N) = \frac{1}{2} x'_N P_N x_N$  in which  $P_N$  is a symmetric positive semi-definite matrix. The following part of the proof is by induction. Assume that the value function,  $V_{k+1}(x_{k+1})$ , at stage  $k+1$  is given by

$$V_{k+1}(x_{k+1}) = \frac{1}{2} x'_{k+1} P_{k+1} x_{k+1} \quad (4.28)$$

in which  $P_{k+1}$  is symmetric positive semi-definite. This assumption implies

$$\begin{aligned} \phi_k(x_k, u_k) &= l_k(x_k, u_k) + V_{k+1}(A'_k x_k + B'_k u_k) \\ &= \frac{1}{2} x'_k Q_k x_k + x'_k M_k u_k + \frac{1}{2} u'_k R_k u_k + \frac{1}{2} (A'_k x_k + B'_k u_k)' P_{k+1} (A'_k x_k + B'_k u_k) \\ &= \frac{1}{2} x'_k (Q_k + A_k P_{k+1} A'_k) x_k + x'_k (M_k + A_k P_{k+1} B'_k) u_k + \frac{1}{2} u'_k (R_k + B_k P_{k+1} B'_k) u_k \end{aligned} \quad (4.29)$$

The partial first and second order derivatives of this function with respect to  $u_k$  are

$$\nabla_{u_k} \phi_k(x_k, u_k) = (M_k + A_k P_{k+1} B'_k)' x_k + (R_k + B_k P_{k+1} B'_k) u_k \quad (4.30)$$

and

$$\nabla_{u_k, u_k}^2 \phi_k(x_k, u_k) = R_k + B_k P_{k+1} B'_k = R_{e,k} \quad (4.31)$$

$R_{e,k}$  is obviously symmetric as  $R_k$  is symmetric and  $P_{k+1}$  is symmetric.  $R_{e,k}$  is also positive definite as  $R_k$  is positive definite by assumption and  $P_{k+1}$  is positive semi-definite. Consequently, the unique global minimizer,  $u_k^*$ , of

$$V_k(x_k) = \min_{u_k} \phi_k(x_k, u_k) \quad (4.32)$$

may be found by the necessary and sufficient condition

$$\nabla_{u_k} \phi_k(x_k, u_k^*) = 0 \quad (4.33)$$

which is equivalent with the expression

$$\nabla_{u_k} \phi_k(x_k, u_k^*) = (M_k + A_k P_{k+1} B'_k)' x_k + (R_k + B_k P_{k+1} B'_k) u_k^* = 0 \quad (4.34)$$



As  $R_{e,k} = R_k + B_k P_{k+1} B'_k$  is symmetric positive definite, it is also non-singular and the unique global minimizer  $u_k^*$  may be expressed as the following function of  $x_k$

$$\begin{aligned} u_k^* &= -(R_k + B_k P_{k+1} B'_k)^{-1} (M_k + A_k P_{k+1} B'_k)' x_k \\ &= K_k x_k \end{aligned} \quad (4.35)$$

in which  $K_k$  is defined by

$$\begin{aligned} K_k &= -(R_k + B_k P_{k+1} B'_k)^{-1} (M_k + A_k P_{k+1} B'_k)' \\ &= -R_{e,k}^{-1} (M_k + A_k P_{k+1} B'_k)' \end{aligned} \quad (4.36)$$

The functional relation giving the optimal control may also be denoted  $u_k^* = \mu_k(x_k) = K_k x_k$  to stress the fact that the optimal control is a function of  $x_k$ . Applying  $u_k^* = K_k x_k$ , the value function  $V_k(x_k)$  may be expressed as

$$\begin{aligned} V_k(x_k) &= \phi_k(x_k, u_k^*) = \phi_k(x_k, K_k x_k) \\ &= \frac{1}{2} x'_k (Q_k + A_k P_{k+1} A'_k) x_k + x'_k (M_k + A_k P_{k+1} B'_k) K_k x_k \\ &\quad + \frac{1}{2} x'_k K'_k (R_k + B_k P_{k+1} B'_k) K_k x_k \\ &= \frac{1}{2} x'_k (Q_k + A_k P_{k+1} A'_k) x_k - x'_k (M_k + A_k P_{k+1} B'_k) R_{e,k}^{-1} (M_k + A_k P_{k+1} B'_k)' x_k \\ &\quad + \frac{1}{2} x_k (M_k + A_k P_{k+1} B'_k) R_{e,k}^{-1} R_{e,k} R_{e,k}^{-1} (M_k + A_k P_{k+1} B'_k)' x_k \\ &= \frac{1}{2} x'_k \left[ Q_k + A_k P_{k+1} A'_k - (M_k + A_k P_{k+1} B'_k) R_{e,k}^{-1} (M_k + A_k P_{k+1} B'_k)' \right] x_k \\ &= \frac{1}{2} x'_k P_k x_k \end{aligned} \quad (4.37)$$

in which

$$\begin{aligned} P_k &= Q_k + A_k P_{k+1} A'_k - (M_k + A_k P_{k+1} B'_k) R_{e,k}^{-1} (M_k + A_k P_{k+1} B'_k)' \\ &= Q_k + A_k P_{k+1} A'_k - (M_k + A_k P_{k+1} B'_k) R_{e,k}^{-1} R_{e,k} R_{e,k}^{-1} (M_k + A_k P_{k+1} B'_k)' \\ &= Q_k + A_k P_{k+1} A'_k - K'_k R_{e,k} K_k \end{aligned} \quad (4.38)$$

$P_k$  is symmetric as  $Q_k$ ,  $P_{k+1}$ , and  $R_{e,k} = R_k + B_k P_{k+1} B'_k$  are symmetric.  $P_{k+1}$  is also positive semi-definite as

$$\begin{aligned} \frac{1}{2} x'_k P_k x_k &= V_k(x_k) \\ &= \min_{u_k} \{l_k(x_k, u_k) + V_{k+1}(A'_k x_k + B'_k u_k)\} \\ &= \min_{u_k} \left\{ \underbrace{\frac{1}{2} \begin{pmatrix} x_k \\ u_k \end{pmatrix}' \begin{pmatrix} Q_k & M_k \\ M'_k & R_k \end{pmatrix} \begin{pmatrix} x_k \\ u_k \end{pmatrix}}_{\geq 0} + \underbrace{\frac{1}{2} (A'_k x_k + B'_k u_k)' P_{k+1} (A'_k x_k + B'_k u_k)}_{\geq 0} \right\} \\ &\geq 0 \end{aligned} \tag{4.39}$$

Consequently, these fact establish the claim that the value function  $V_k(x_k)$  is

$$V_k(x_k) = \frac{1}{2} x'_k P_k x_k \quad k = 0, 1, \dots, N \tag{4.40}$$

in which  $P_k$  is symmetric positive semi-definite. Hence, the optimal value,  $\phi^*$ , is

$$\begin{aligned} \phi^* &= \min_{\{x_{k+1}, u_k\}_{k=0}^{N-1}} \left\{ \sum_{k=0}^{N-1} l_k(x_k, u_k) + l_N(x_N) : x_{k+1} = A'_k x_k + B'_k u_k, k = 0, 1, \dots, N-1 \right\} \\ &= V_0(x_0) = \frac{1}{2} x'_0 P_0 x_0 \end{aligned} \tag{4.41}$$

and the global minimizer  $\{x_{k+1}^*, u_k^*\}_{k=0}^{N-1}$  is

$$u_k^* = K_k x_k^* \tag{4.42a}$$

$$x_{k+1} = A'_k x_k^* + B'_k u_k^* \tag{4.42b}$$

for  $k = 0, 1, \dots, N-1$  and  $x_0^* = x_0$ . By the deduction it is evident that  $P_N$  is given and  $R_{e,k}$ ,  $K_k$ , and  $P_k$  are obtained iteratively by

$$R_{e,k} = R_k + B_k P_{k+1} B'_k \tag{4.43a}$$

$$K_k = -R_{e,k}^{-1} (M_k + A_k P_{k+1} B'_k)' \tag{4.43b}$$

$$P_k = Q_k + A_k P_{k+1} A_k - K'_k R_{e,k} K_k \tag{4.43c}$$

for  $k = N-1, N-2, \dots, 1, 0$ .  $\square$

#### Remark 4.2.6

The assumption that  $\{R_k\}_{k=0}^{N-1}$  are positive definite is sufficient for the result in proposition 4.2.5 being a solution of (4.15). However, it is not necessary. From the preceding proof, it is obvious that a necessary and sufficient condition for  $\{x_{k+1}, u_k\}_{k=0}^{N-1}$  being a unique minimizer of (4.15) is that  $\{R_{e,k} = R_k + B_k P_{k+1} B'_k\}_{k=0}^{N-1}$  is positive definite.

**Remark 4.2.7**

The expressions for  $\{K_k\}_{k=0}^{N-1}$  and  $\{P_k\}_{k=0}^{N-1}$  may also be formulated as

$$K_k = -(R_k + B_k P_{k+1} B'_k)^{-1} (M_k + A_k P_{k+1} B'_k)' \quad (4.44a)$$

$$P_k = Q_k + A_k P_{k+1} A'_k - (M_k + A_k P_{k+1} B'_k) (R_k + B_k P_{k+1} B'_k)^{-1} (M_k + A_k P_{k+1} B'_k)' \quad (4.44b)$$

**Corollary 4.2.8**

Let assumptions 4.2.3 and 4.2.4 be satisfied. Then the matrices

$$P_k = Q_k + A_k P_{k+1} A'_k - (M_k + A_k P_{k+1} B'_k) (R_k + B_k P_{k+1} B'_k)^{-1} (M_k + A_k P_{k+1} B'_k)' \quad (4.45)$$

in the sequence of matrices  $\{P_k\}_{k=0}^{N-1}$ , are all symmetric positive semi-definite.

*Proof.* Follows directly from the proof of proposition 4.2.5.  $\square$

## 4.3 Extended Linear-Quadratic Optimal Control

In this section, an algorithm for solving the extended linear-quadratic optimal control problem is established. The extended linear-quadratic optimal control problem is established in the following problem definition.

**Problem 4.3.1 (Extended Linear-Quadratic Optimal Control)**

The extended linear-quadratic optimal control problem consists of solving the quadratic program

$$\min_{\{x_{k+1}, u_k\}_{k=0}^{N-1}} \phi = \sum_{k=0}^{N-1} l_k(x_k, u_k) + l_N(x_N) \quad (4.46a)$$

$$s.t. \quad x_{k+1} = A'_k x_k + B'_k u_k + b_k \quad k = 0, 1, \dots, N-1 \quad (4.46b)$$

with the stage costs given by

$$l_k(x_k, u_k) = \frac{1}{2} x'_k Q_k x_k + x'_k M_k u_k + \frac{1}{2} u'_k R_k u_k + q'_k x_k + r'_k u_k + f_k \quad k = 0, 1, \dots, N-1 \quad (4.47a)$$

$$l_N(x_N) = \frac{1}{2} x'_N P_N x_N + p'_N x_N + \gamma_N \quad (4.47b)$$

In (4.46),  $x_0$  is a parameter and not a decision variable. The optimal solution consists of the minimizer  $(x^*, u^*) = \{x_{k+1}^*, u_k^*\}_{k=0}^{N-1}$  and the optimal value  $\phi^* = \phi^*(x^*, u^*)$ .

**Remark 4.3.2**

The stage costs (4.47a) may also be expressed as

$$\begin{aligned} l_k(x_k, u_k) &= \frac{1}{2} x'_k Q_k x_k + x'_k M_k u_k + \frac{1}{2} u'_k R_k u_k + q'_k x_k + r'_k u_k + f_k \\ &= \frac{1}{2} \begin{pmatrix} x_k \\ u_k \end{pmatrix}' \begin{pmatrix} Q_k & M_k \\ M'_k & R_k \end{pmatrix} \begin{pmatrix} x_k \\ u_k \end{pmatrix} + \begin{pmatrix} q_k \\ r_k \end{pmatrix}' \begin{pmatrix} x_k \\ u_k \end{pmatrix} + f_k \end{aligned} \quad (4.48)$$

The important extensions of the extended linear-quadratic optimal control problem compared to the linear quadratic optimal control problem are the linear terms in the stage costs of the objective function as well as the affine term in the dynamic equations. These terms have usually been neglected in the classic treatment of the linear-quadratic optimal control problem. However, these terms are important in the solution of the nonlinear optimal control problem as well as in the solution of constrained problems with a steady-state residing on the boundary of the feasible region.

The assumptions stated next are used to guarantee that the extended linear-quadratic optimal control problem is strictly convex such that the resulting optimal minimizer is unique.

**Assumption 4.3.3 (Symmetric Positive Semi-Definiteness)**

All matrices

$$\begin{pmatrix} Q_k & M_k \\ M'_k & R_k \end{pmatrix} \quad (4.49)$$

in the sequence of matrices

$$\left\{ \begin{pmatrix} Q_k & M_k \\ M'_k & R_k \end{pmatrix} \right\}_{k=0}^{N-1} \quad (4.50)$$

are symmetric positive semi-definite.  $P_N$  is symmetric positive semi-definite.

**Assumption 4.3.4 (Sufficient Uniqueness and Optimality Condition)**

All matrices,  $R_k$ , in the sequence of matrices  $\{R_k\}_{k=0}^{N-1}$  are positive definite.

The solution of the extended linear-quadratic optimal control problem is stated in the following proposition.

**Proposition 4.3.5 (Solution of the Extended Linear-Quadratic Optimal Control Problem)**

Consider the linear-quadratic optimal control problem (4.46). Let assumptions 4.3.3 and 4.3.4 be satisfied.

Let the sequence of matrices  $\{R_{e,k}, K_k, P_k\}_{k=0}^{N-1}$  be defined by

$$R_{e,k} = R_k + B_k P_{k+1} B'_k \quad (4.51a)$$

$$K_k = -R_{e,k}^{-1} (M_k + A_k P_{k+1} B'_k)' \quad (4.51b)$$

$$P_k = Q_k + A_k P_{k+1} A'_k - K'_k R_{e,k} K_k \quad (4.51c)$$

and let the vectors  $\{c_k, d_k, a_k, p_k\}_{k=0}^{N-1}$  be defined by

$$c_k = P_{k+1} b_k + p_{k+1} \quad (4.52a)$$

$$d_k = r_k + B_k c_k \quad (4.52b)$$

$$a_k = -R_{e,k}^{-1} d_k \quad (4.52c)$$

$$p_k = q_k + A_k c_k + K'_k d_k \quad (4.52d)$$

Let the sequence of scalars  $\{\gamma_k\}_{k=0}^{N-1}$  be defined by

$$\gamma_k = \gamma_{k+1} + f_k + p'_{k+1} b_k + \frac{1}{2} b'_k P_{k+1} b_k + \frac{1}{2} d'_k a_k \quad (4.53)$$

Let  $x_0^* = x_0$ . Then the unique global minimizer  $\{x_{k+1}^*, u_k^*\}_{k=0}^{N-1}$  of (4.46) may be obtained by the iteration

$$u_k^* = K_k x_k^* + a_k \quad (4.54a)$$

$$x_{k+1}^* = A_k' x_k^* + B_k' u_k^* + b_k \quad (4.54b)$$

The corresponding optimal value,  $\phi^*$ , of (4.46) may be computed by

$$\phi^* = \frac{1}{2} x_0' P_0 x_0 + p_0' x_0 + \gamma_0 \quad (4.55)$$

*Proof.* Let  $V_k(x_k)$  denote the value function defined by

$$V_k(x_k) = \min_{\{x_{i+1}, u_i\}_{i=k}^{N-1}} \left\{ \sum_{i=k}^{N-1} l_i(x_i, u_i) + l_N(x_N) : x_{i+1} = A_i' x_i + B_i' u_i + b_i, i = k, k+1, \dots, N-1 \right\} \quad (4.56)$$

Note that  $V_0(x_0)$  corresponds to the optimal value of (4.46).  $V_k(x_k)$  corresponds to the optimal value of an extended linear-quadratic optimal control problem starting at stage  $k$  with state  $x_k$  given. By the above definition of the value function, it is clear that

$$V_N(x_N) = l_N(x_N) = \frac{1}{2} x_N' P_N x_N + p_N' x_N + \gamma_N \quad (4.57)$$

in which  $P_N$  by assumption is a symmetric positive semi-definite matrix. By Bellman's principle of optimality and the dynamic programming algorithm (c.f. Bertsekas, 1995a), the value function  $V_k(x_k)$  may be expressed recursively as

$$\begin{aligned} V_k(x_k) &= \min_{x_{k+1}, u_k} \{l_k(x_k, u_k) + V_{k+1}(x_{k+1}) : x_{k+1} = A_k' x_k + B_k' u_k + b_k\} \\ &= \min_{u_k} l_k(x_k, u_k) + V_{k+1}(A_k' x_k + B_k' u_k + b_k) \\ &= \min_{u_k} \phi_k(x_k, u_k) \end{aligned} \quad (4.58)$$

in which the function  $\phi_k(x_k, u_k)$  is defined by

$$\phi_k(x_k, u_k) = l_k(x_k, u_k) + V_{k+1}(A_k' x_k + B_k' u_k + b_k) \quad (4.59)$$

To obtain the result claimed in the proposition, we want to show that

$$V_k(x_k) = \frac{1}{2} x_k' P_k x_k + p_k' x_k + \gamma_k \quad (4.60)$$

in which  $P_k$  is a symmetric positive semi-definite matrix. Hence, we want to show that the value function  $V_k(x_k)$  is a convex quadratic function. Observe that this expression is valid for  $k = N$  as

$$V_N(x_N) = l_N(x_N) = \frac{1}{2} x_N' P_N x_N + p_N' x_N + \gamma_N \quad (4.61)$$

and  $P_N$  is a symmetric positive semi-definite matrix by assumption.  $V_k(x_k)$  is proved to be a convex quadratic function by induction. Assume that

$$V_{k+1}(x_{k+1}) = \frac{1}{2} x'_{k+1} P_{k+1} x_{k+1} + p'_{k+1} x_{k+1} + \gamma_{k+1} \quad (4.62)$$

in which  $P_{k+1}$  is a symmetric positive semi-definite matrix. Then

$$\begin{aligned} V_{k+1}(A'_k x_k + B'_k u_k + b_k) &= \frac{1}{2} (A'_k x_k + B'_k u_k + b_k)' P_{k+1} (A'_k x_k + B'_k u_k + b_k) \\ &\quad + p'_{k+1} (A'_k x_k + B'_k u_k + b_k) + \gamma_{k+1} \\ &= \frac{1}{2} x'_k A_k P_{k+1} A'_k x_k + x'_k A_k P_{k+1} B'_k u_k + \frac{1}{2} u'_k B_k P_{k+1} B'_k u_k \\ &\quad + [A_k (P_{k+1} b_k + p_{k+1})]' x_k + [B_k (P_{k+1} b_k + p_{k+1})]' u_k \\ &\quad + \left( \gamma_{k+1} + p'_{k+1} b_k + \frac{1}{2} b'_k P_{k+1} b_k \right) \end{aligned} \quad (4.63)$$

and

$$\begin{aligned} \phi_k(x_k, u_k) &= l_k(x_k, u_k) + V_{k+1}(A'_k x_k + B'_k u_k + b_k) \\ &= \frac{1}{2} x'_k (Q_k + A_k P_{k+1} A'_k) x_k + x'_k (M_k + A_k P_{k+1} B'_k) u_k + \frac{1}{2} u'_k (R_k + B_k P_{k+1} B'_k) u_k \\ &\quad + [q_k + A_k (P_{k+1} b_k + p_{k+1})]' x_k + [r_k + B_k (P_{k+1} b_k + p_{k+1})]' u_k \\ &\quad + \left( f_k + \gamma_{k+1} + p'_{k+1} b_k + \frac{1}{2} b'_k P_{k+1} b_k \right) \end{aligned} \quad (4.64)$$

The value function  $V_k(x_k)$  satisfies

$$V_k(x_k) = \min_{u_k} \phi_k(x_k, u_k) \quad (4.65)$$

A necessary and sufficient condition for  $u_k^*$  being a global minimizer of this expression is

$$\nabla_{u_k} \phi_k(x_k, u_k^*) = 0 \quad (4.66)$$

as the Hessian

$$\nabla_{u_k, u_k}^2 \phi_k(x_k, u_k) = R_k + B_k P_{k+1} B'_k = R_{e,k} \quad (4.67)$$

is positive definite.  $R_{e,k} = R_k + B_k P_{k+1} B'_k$  is symmetric positive definite as  $R_k$  is symmetric positive definite and  $P_{k+1}$  is symmetric positive semi-definite. The necessary and sufficient optimality condition yields

$$\nabla_{u_k} \phi_k(x_k, u_k^*) = (M_k + A_k P_{k+1} B'_k)' x_k + (R_k + B_k P_{k+1} B'_k) u_k^* + [r_k + B_k (P_{k+1} b_k + p_{k+1})] = 0 \quad (4.68)$$

$R_{e,k} = R_k + B_k P_{k+1} B'_k$  is positive definite and therefore also non-singular. This implies that the unique global minimizer  $u_k^*$  may be expressed as

$$\begin{aligned} u_k^* &= -(R_k + B_k P_{k+1} B'_k)^{-1} (M_k + A_k P_{k+1} B'_k)' x_k \\ &\quad - (R_k + B_k P_{k+1} B'_k)^{-1} (r_k + B_k (P_{k+1} b_k + p_{k+1})) \\ &= K_k x_k + a_k \end{aligned} \quad (4.69)$$

The gain  $K_k$  and affine term  $a_k$  are defined by

$$K_k = -(R_k + B_k P_{k+1} B'_k)^{-1} (M_k + A_k P_{k+1} B'_k)' = -R_{e,k}^{-1} (M_k + A_k P_{k+1} B'_k)' \quad (4.70a)$$

$$a_k = -(R_k + B_k P_{k+1} B'_k)^{-1} (r_k + B_k (P_{k+1} b_k + p_{k+1})) = -R_{e,k}^{-1} d_k \quad (4.70b)$$

in which

$$d_k = r_k + B_k (P_{k+1} b_k + p_{k+1}) = r_k + B_k c_k \quad (4.71)$$

and

$$c_k = P_{k+1} b_k + p_{k+1} \quad (4.72)$$

Application of the result,  $u_k^* = K_k x_k + a_k$ , gives the following expression for the value function

$$\begin{aligned} V_k(x_k) &= \phi_k(x_k, u_k^*) = \phi_k(x_k, K_k x_k + a_k) \\ &= \frac{1}{2} x_k' (Q_k + A_k P_{k+1} A'_k) x_k + x_k' (M_k + A_k P_{k+1} B'_k) (K_k x_k + a_k) \\ &\quad + \frac{1}{2} (K_k x_k + a_k)' (R_k + B_k P_{k+1} B'_k) (K_k x_k + a_k) \\ &\quad + [q_k + A_k (P_{k+1} b_k + p_{k+1})]' x_k + [r_k + B_k (P_{k+1} b_k + p_{k+1})]' (K_k x_k + a_k) \\ &\quad + \left( f_k + \gamma_{k+1} + p'_{k+1} b_k + \frac{1}{2} b'_k P_{k+1} b_k \right) \\ &= \frac{1}{2} x_k' (Q_k + A_k P_{k+1} A'_k + 2(M_k + A_k P_{k+1} B'_k) K_k + K'_k (R_k + B_k P_{k+1} B'_k) K_k) x_k \\ &\quad + [(M_k + A_k P_{k+1} B'_k) a_k + K'_k (R_k + B_k P_{k+1} B'_k) a_k + (q_k + A_k (P_{k+1} b_k + p_{k+1})) \\ &\quad + K'_k (r_k + B_k (P_{k+1} b_k + p_{k+1}))]' x_k \\ &\quad + \left( f_k + \gamma_{k+1} + p'_{k+1} b_k + \frac{1}{2} b'_k P_{k+1} b_k \right. \\ &\quad \left. + [r_k + B_k (P_{k+1} b_k + p_{k+1})]' a_k + \frac{1}{2} a'_k (R_k + B_k P_{k+1} B'_k) a_k \right) \\ &= \frac{1}{2} x_k' P_k x_k + p'_k x_k + \gamma_k \end{aligned} \quad (4.73)$$

The quadratic weight matrix  $P_k$  is defined by

$$\begin{aligned}
P_k &= Q_k + A_k P_{k+1} A'_k + 2(M_k + A_k P_{k+1} B'_k) K_k + K'_k (R_k + B_k P_{k+1} B'_k) K_k \\
&= Q_k + A_k P_{k+1} A'_k - 2(M_k + A_k P_{k+1} B'_k) R_{e,k}^{-1} (M_k + A_k P_{k+1} B'_k)' \\
&\quad + (M_k + A_k P_{k+1} B'_k) R_{e,k}^{-1} R_{e,k} R_{e,k}^{-1} (M_k + A_k P_{k+1} B'_k)' \\
&= Q_k + A_k P_{k+1} A'_k - (M_k + A_k P_{k+1} B'_k) R_{e,k}^{-1} (M_k + A_k P_{k+1} B'_k)' \\
&= Q_k + A_k P_{k+1} A'_k - (M_k + A_k P_{k+1} B'_k) R_{e,k}^{-1} R_{e,k} R_{e,k}^{-1} (M_k + A_k P_{k+1} B'_k)' \\
&= Q_k + A_k P_{k+1} A'_k - K'_k R_{e,k} K_k
\end{aligned} \tag{4.74}$$

and the linear weight matrix  $p_k$  is defined by

$$\begin{aligned}
p_k &= (M_k + A_k P_{k+1} B'_k) a_k + K'_k (R_k + B_k P_{k+1} B'_k) a_k + (q_k + A_k (P_{k+1} b_k + p_{k+1})) \\
&\quad + K'_k (r_k + B_k (P_{k+1} b_k + p_{k+1})) \\
&= \underbrace{(M_k + A_k P_{k+1} B'_k) a_k - (M_k + A_k P_{k+1} B'_k) R_{e,k}^{-1} R_{e,k} a_k}_{=0} \\
&\quad + q_k + A_k (P_{k+1} b_k + p_{k+1}) + K'_k (r_k + B_k (P_{k+1} b_k + p_{k+1})) \\
&= q_k + A_k \underbrace{(P_{k+1} b_k + p_{k+1})}_{=c_k} + K'_k \underbrace{(r_k + B_k (P_{k+1} b_k + p_{k+1}))}_{=d_k} \\
&= q_k + A_k c_k + K'_k d_k
\end{aligned} \tag{4.75}$$

Similarly, the scalar  $\gamma_k$  is defined by

$$\begin{aligned}
\gamma_k &= f_k + \gamma_{k+1} + p'_{k+1} b_k + \frac{1}{2} b'_k P_{k+1} b_k \\
&\quad + [r_k + B_k (P_{k+1} b_k + p_{k+1})]' a_k + \frac{1}{2} a'_k (R_k + B_k P_{k+1} B'_k) a_k \\
&= f_k + \gamma_{k+1} + p'_{k+1} b_k + \frac{1}{2} b'_k P_{k+1} b_k + d'_k a_k - \frac{1}{2} a'_k R_{e,k} R_{e,k}^{-1} d_k \\
&= f_k + \gamma_{k+1} + p'_{k+1} b_k + \frac{1}{2} b'_k P_{k+1} b_k + \frac{1}{2} d'_k a_k
\end{aligned} \tag{4.76}$$

The matrices  $P_k$  i  $\{P_k\}_{k=0}^{N-1}$  are symmetric positive definite according to corollary 4.2.8. Consequently, it has been established that the value function  $V_k(x_k)$  is

$$V_k(x_k) = \frac{1}{2} x'_k P_k x_k + p'_k x_k + \gamma_k \tag{4.77}$$



for  $k = 0, 1, \dots, N$  and  $P_k$  is symmetric positive semi-definite. The optimal value  $\phi^*$  is

$$\begin{aligned} \phi^* &= \min_{\{x_{k+1}, u_k\}_{k=0}^{N-1}} \left\{ \sum_{k=0}^{N-1} l_k(x_k, u_k) + l_N(x_N) : x_{k+1} = A'_k x_k + B'_k u_k + b_k, k = 0, 1, \dots, N-1 \right\} \\ &= V_0(x_0) = \frac{1}{2} x'_0 P_0 x_0 + p'_0 x_0 + \gamma_0 \end{aligned} \quad (4.78)$$

and the global minimizer  $\{x_{k+1}^*, u_k^*\}_{k=0}^{N-1}$  is computed by

$$u_k^* = K_k x_k^* + a_k \quad (4.79a)$$

$$x_{k+1}^* = A'_k x_k^* + B'_k u_k^* + b_k \quad (4.79b)$$

for  $k = 0, 1, \dots, N-1$ .  $x_0^* = x_0$  by convention.

By the deduction above, it has been established that

$$R_{e,k} = R_k + B_k P_{k+1} B'_k \quad (4.80a)$$

$$K_k = -R_{e,k}^{-1} (M_k + A_k P_{k+1} B'_k)' \quad (4.80b)$$

$$P_{k+1} = Q_k + A_k P_{k+1} A'_k - K'_k R_{e,k} K_k \quad (4.80c)$$

and

$$c_k = P_{k+1} b_k + p_{k+1} \quad (4.81a)$$

$$d_k = r_k + B_k c_k \quad (4.81b)$$

$$a_k = -R_{e,k}^{-1} d_k \quad (4.81c)$$

$$p_k = q_k + A_k c_k + K'_k d_k \quad (4.81d)$$

Furthermore

$$\gamma_k = f_k + \gamma_{k+1} + p'_{k+1} b_k + \frac{1}{2} b'_k P_{k+1} b_k + \frac{1}{2} d'_k a_k \quad (4.82)$$

□

**Remark 4.3.6**

Note that recursion for  $p_k$  may be expressed as

$$p_k = q_k + A_k (P_{k+1} b_k + p_{k+1}) + K'_k (r_k + B_k (P_{k+1} b_k + p_{k+1})) \quad (4.83)$$

By this formulation, it is evident that  $p_k$  depends on  $p_{k+1}$ .

As has just been established, the optimal control inputs for the extended linear-quadratic optimal control problem are affine functions of the states. In contrast, the optimal control inputs for the linear-quadratic optimal control problem are linear functions of the states.

### 4.3.1 Algorithms

The efficient computation of the solution of the extended linear-quadratic optimal control problem is facilitated by the next corollary.

**Corollary 4.3.7**

Let assumptions 4.3.3 and 4.3.4 be satisfied. Let  $\{R_{e,k}, K_k, P_k\}_{k=0}^{N-1}$  be defined as in proposition 4.3.5. Let  $\{c_k, d_k, a_k, p_k\}_{k=0}^{N-1}$  be defined as in proposition 4.3.5.

Then  $R_{e,k}$  is positive definite and has the Cholesky factorization

$$R_{e,k} = L_k L_k' \quad (4.84)$$

in which  $L_k$  is a non-singular lower triangular matrix.

Further, define

$$Y_k = (M_k + A_k P_{k+1} B_k')' \quad (4.85)$$

and

$$Z_k = L_k^{-1} Y_k \quad (4.86)$$

$$z_k = L_k^{-1} d_k \quad (4.87)$$

Then

$$P_k = Q_k + A_k P_{k+1} A_k' - Z_k' Z_k \quad (4.88)$$

$$p_k = q_k + A_k c_k - Z_k' z_k \quad (4.89)$$

and  $u_k = K_k x_k + a_k$  may be computed according to

$$u_k = -(L_k')^{-1} (Z_k x_k + z_k) \quad (4.90)$$

*Proof.* By assumption 4.3.4,  $R_k$  is positive definite. In the proof of proposition 4.3.5 it has been established that  $P_{k+1}$  is positive semi-definite for all  $k = 0, 1, \dots, N-1$ . Therefore,  $R_{e,k} = R_k + B_k P_{k+1} B_k'$  is positive definite and has the Cholesky factorization

$$R_{e,k} = L_k L_k' \quad (4.91)$$

The matrix  $Y_k$  is defined as

$$Y_k = (M_k + A_k P_{k+1} B_k')' \quad (4.92)$$

and the matrix  $Z_k$  is defined as

$$Z_k = -L_k^{-1} Y_k = -L_k^{-1} (M_k + A_k P_{k+1} B_k')' \quad (4.93)$$

Then

$$\begin{aligned} K_k &= -R_{e,k}^{-1} (M_k + A_k P_{k+1} B_k')' \\ &= -(L_k L_k')^{-1} Y_k \\ &= -(L_k')^{-1} L_k^{-1} Y_k \\ &= -(L_k')^{-1} Z_k \end{aligned} \quad (4.94)$$

Applying the deduced relations for  $K_k$  and  $R_{e,k}$  in the recursion for  $P_k$  yields

$$\begin{aligned}
 P_k &= Q_k + A_k P_{k+1} A'_k - K'_k R_{e,k} K_k \\
 &= Q_k + A_k P_{k+1} A'_k - [-(L'_k)^{-1} Z_k]' L_k L'_k [-(L'_k)^{-1} Z_k] \\
 &= Q_k + A_k P_{k+1} A'_k - Z'_k L_k^{-1} L_k L'_k (L'_k)^{-1} Z_k \\
 &= Q_k + A_k P_{k+1} A'_k - Z'_k Z_k
 \end{aligned} \tag{4.95}$$

Define the vector  $z_k$  as

$$z_k = L_k^{-1} d_k \tag{4.96}$$

in which  $d_k$  is computed according to proposition 4.3.5. Then the recursion for  $p_k$  may be expressed as

$$\begin{aligned}
 p_k &= q_k + A_k c_k + K'_k d_k \\
 &= q_k + A_k c_k + [-(L'_k)^{-1} Z_k]' d_k \\
 &= q_k + A_k c_k - Z'_k L_k^{-1} d_k \\
 &= q_k + A_k c_k - Z'_k z_k
 \end{aligned} \tag{4.97}$$

The expression for  $a_k$  given by proposition 4.3.5 may be developed to

$$\begin{aligned}
 a_k &= -R_{e,k}^{-1} d_k \\
 &= -(L'_k)^{-1} L_k^{-1} d_k \\
 &= -(L'_k)^{-1} z_k
 \end{aligned} \tag{4.98}$$

such that  $u_k$  may be computed by

$$\begin{aligned}
 u_k &= K_k x_k + a_k \\
 &= -(L'_k)^{-1} Z_k x_k - (L'_k)^{-1} z_k \\
 &= -(L'_k)^{-1} (Z_k x_k + z_k)
 \end{aligned} \tag{4.99}$$

□

Algorithm 1 provides the major steps in factorizing and solving the extended linear quadratic optimal control problem (4.46). This algorithm factorizes the KKT-matrix according to the Riccati recursion and computes the minimizer  $\{x_{k+1}, u_k\}_{k=0}^{N-1}$  as well as the optimal value  $\phi$  of (4.46). The major computational step in this algorithm is typically the computation of the term  $A_k P_{k+1} A'_k$  by computation of  $S = A_k P_{k+1}$  and  $S A'_k$ . In this computation it is important to utilize that the matrices  $P_k$  for  $k = 0, 1, \dots, N$  are symmetric. Then only half the entries of  $P_k$  need to be computed. Similarly, it may be utilized that the matrices  $R_{e,k} = R_k + B_k P_{k+1} B'_k$  are symmetric for  $k = 0, 1, \dots, N - 1$ .

**Remark 4.3.8**

For cases in which symmetry of  $P$  is not utilized the algorithm may diverge due to numerical instability. Empirical evidence suggest that algorithm is stabilized by

---

**Algorithm 1** Solution of the extended linear-quadratic optimal control problem.

---

**Require:**  $N$ ,  $(P_N, p_N, \gamma_N)$ ,  $\{Q_k, M_k, R_k, q_k, f_k, r_k, A_k, B_k, b_k\}_{k=0}^{N-1}$ , and  $x_0$ .

Assign  $P \leftarrow P_N$ ,  $p \leftarrow p_N$ , and  $\gamma \leftarrow \gamma_N$ .

**for**  $k = N - 1 : -1 : 0$  **do**

    Compute the temporary matrices and vectors

$$R_e = R_k + B_k P B_k' \quad (4.100a)$$

$$S = A_k P \quad (4.100b)$$

$$Y = (M_k + S B_k')' \quad (4.100c)$$

$$s = P b_k \quad (4.100d)$$

$$c = s + p \quad (4.100e)$$

$$d = r_k + B_k c \quad (4.100f)$$

    Cholesky factorize  $R_e$

$$R_e = L_k L_k' \quad (4.101)$$

    Compute  $Z_k$  and  $z_k$  by solving

$$L_k Z_k = Y \quad (4.102a)$$

$$L_k z_k = d \quad (4.102b)$$

    Update  $P$ ,  $\gamma$ , and  $p$  by

$$P \leftarrow Q_k + S A_k' - Z_k' Z_k \quad (4.103a)$$

$$\gamma \leftarrow \gamma + f_k + p' b_k + \frac{1}{2} s' b_k - \frac{1}{2} z_k' z_k \quad (4.103b)$$

$$p \leftarrow q_k + A_k c - Z_k' z_k \quad (4.103c)$$

**end for**

Compute the optimal value by

$$\phi = \frac{1}{2} x_0' P x_0 + p' x_0 + \gamma \quad (4.104)$$

**for**  $k = 0 : 1 : N - 1$  **do**

    Compute

$$y = Z_k x_k + z_k \quad (4.105)$$

    and solve the linear system of equations

$$L_k' u_k = -y \quad (4.106)$$

    for  $u_k$ .

    Compute

$$x_{k+1} = A_k' x_k + B_k' u_k + b_k \quad (4.107)$$

**end for**

Return  $\{x_{k+1}, u_k\}_{k=0}^{N-1}$  and  $\phi$ .

---

forcing symmetry of  $P$ . Hence, in the computation of the entire matrix  $P$ , it should be computed according to

$$P \leftarrow Q_k + SA'_k - Z'_k Z_k \quad (4.108a)$$

$$P \leftarrow \frac{1}{2} (P + P') \quad (4.108b)$$

In some practical applications, the extended linear-quadratic optimal control problem must be solved for different values of the data  $(x_0, \{q_k, r_k, b_k, f_k\}_{k=0}^{N-1}, \{p_N, \gamma_N\})$  but unaltered values of the data  $(\{Q_k, M_k, R_k, A_k, B_k\}_{k=0}^{N-1}, P_N)$ . In such situations, it is inefficient to call algorithm 1 repeatedly to solve these problems. Instead, one factorization, i.e. computation of  $\{L_k, Z_k, P_k\}_{k=0}^{N-1}$ , may be computed for the unaltered part of the data, i.e.  $(\{Q_k, M_k, R_k, A_k, B_k\}_{k=0}^{N-1}, P_N)$ . This factorization is stated in algorithm 2. The computation of  $\{L_k, Z_k, P_k\}_{k=0}^{N-1}$  is the major cost of solving the extended linear-quadratic optimal control problem. Given the factorization  $\{L_k, Z_k, P_k\}_{k=0}^{N-1}$ , the extended linear-quadratic optimal control problem may be solved for each data set  $(x_0, \{q_k, r_k, b_k, f_k\}_{k=0}^{N-1}, \{p_N, \gamma_N\})$  by application of algorithm 3. Compared to simultaneous factorization and solution of the extended linear-quadratic optimal control problem (i.e. algorithm 1), the sequential factorization and solution of the extended linear-quadratic optimal control problem (i.e. algorithms 2 and 3) requires storage of the entire sequence of matrices constituting the factorization, i.e.  $\{L_k, Z_k, P_k\}_{k=0}^{N-1}$ .

The principles applied in developing an algorithm for factorization and an algorithm for solution may be specialized even further for cases in which  $x_0$  is the only changing data. This may for instance be the case in linear time-invariant model predictive control applications.

## 4.4 Equality Constrained Quadratic Program

This section introduces the general equality constrained quadratic program and describes briefly a method for its solution. The solution, in form of the optimal solution-Lagrange multiplier pair, is obtained by solving the KKT-system corresponding to the first order necessary and sufficient optimality conditions. The computation of the minimizer and associated Lagrange multipliers of an equality constrained quadratic program is then used as a framework for solution of the extended linear-quadratic optimal control problem.

We formally define the equality constrained convex quadratic program by the following problem formulation.

### Problem 4.4.1 (Equality Constrained Convex Quadratic Program)

Let the matrix  $G \in \mathbb{R}^{n \times n}$  be symmetric positive semi-definite. Let  $g \in \mathbb{R}^n$  and  $\rho \in \mathbb{R}$ . Let  $A \in \mathbb{R}^{n \times m}$  and  $b \in \mathbb{R}^m$ .

---

**Algorithm 2** Factorization for the extended linear-quadratic optimal control problem.

---

**Require:**  $N$ ,  $P_N$ , and  $\{Q_k, M_k, R_k, A_k, B_k\}_{k=0}^{N-1}$ .

**for**  $k = N - 1 : -1 : 0$  **do**

    Compute the temporary matrices

$$R_e = R_k + B_k P_{k+1} B_k' \quad (4.109a)$$

$$S = A_k P_{k+1} \quad (4.109b)$$

$$Y = (M_k + S B_k')' \quad (4.109c)$$

    Cholesky factorize  $R_e$

$$R_e = L_k L_k' \quad (4.110)$$

    and compute  $Z_k$  by solving

$$L_k Z_k = Y \quad (4.111)$$

    Compute

$$P_k = Q_k + S A_k' - Z_k' Z_k \quad (4.112)$$

**end for**

Return  $\{P_k, L_k, Z_k\}_{k=0}^{N-1}$ .

---

Then the equality constrained convex quadratic program is

$$\min_{y \in \mathbb{R}^n} \quad \phi = \frac{1}{2} y' G y + g' y + \rho \quad (4.120a)$$

$$s.t. \quad A' y = b \quad (4.120b)$$

The solution of this program is the minimizer  $y^*$  and the optimal value  $\phi^*$ .

The solution of the equality constrained quadratic program (4.120) is obtained using a Lagrange multiplier algorithm. This method employs the Lagrange function of (4.120) and derives the first order necessary and sufficient conditions in terms of this Lagrange function.

**Definition 4.4.2 (Lagrangian and Lagrange Multipliers)**

Let  $y \in \mathbb{R}^n$  and  $\pi \in \mathbb{R}^m$ . Let  $\mathcal{L} : \mathbb{R}^n \times \mathbb{R}^m \mapsto \mathbb{R}$  be defined as

$$\mathcal{L}(y, \pi) = \frac{1}{2} y' G y + g' y + \rho - \pi' (A' y - b) \quad (4.121)$$

$\mathcal{L}$  is the Lagrangian function of (4.120) and  $\pi$  are called the Lagrange multipliers.

**Proposition 4.4.3 (Necessary and Sufficient Optimality Conditions)**

Let  $y \in \mathbb{R}^n$ . Let  $\pi \in \mathbb{R}^m$ . Define the Lagrangian function  $\mathcal{L} : \mathbb{R}^n \times \mathbb{R}^m \mapsto \mathbb{R}$  of (4.120) according to (4.121). Then  $y$  is a global minimizer of the equality constrained convex

**Algorithm 3** Solve a factorized extended linear-quadratic optimal control problem.

**Require:**  $N$ ,  $(P_N, p_N, \gamma_N)$ ,  $\{Q_k, M_k, R_k, q_k, r_k, f_k, A_k, B_k, b_k\}_{k=0}^{N-1}$ ,  $x_0$ , and  $\{P_k, L_k, Z_k\}_{k=0}^{N-1}$ .

Assign  $p \leftarrow p_N$  and  $\gamma \leftarrow \gamma_N$ .

**for**  $k = N - 1 : -1 : 0$  **do**

    Compute the temporary vectors

$$s = P_{k+1}b_k \quad (4.113a)$$

$$c = s + p \quad (4.113b)$$

$$d = r_k + B_k c \quad (4.113c)$$

    Solve the lower triangular system of equations

$$L_k z_k = d \quad (4.114)$$

    for  $z_k$ .

    Update  $\gamma$  and  $p$  by the expressions

$$\gamma \leftarrow \gamma + f_k + p'b_k + \frac{1}{2}s'b_k - \frac{1}{2}z'_k z_k \quad (4.115a)$$

$$p \leftarrow q_k + A_k c - Z'_k z_k \quad (4.115b)$$

**end for**

Compute the optimal value

$$\phi = \frac{1}{2}x'_0 P_0 x_0 + p'_0 x_0 + \gamma \quad (4.116)$$

**for**  $k = 0 : 1 : N - 1$  **do**

    Compute

$$y = Z_k x_k + z_k \quad (4.117)$$

    and solve the upper triangular system of equations

$$L'_k u_k = -y \quad (4.118)$$

    for  $u_k$ .

    Compute

$$x_{k+1} = A'_k x_k + B'_k u_k + b_k \quad (4.119)$$

**end for**

Return  $\{x_{k+1}, u_k\}_{k=0}^{N-1}$  and  $\phi$ .

quadratic program (4.120) if and only if

$$\nabla_y \mathcal{L}(y, \pi) = Gy + g - A\pi = 0 \quad (4.122a)$$

$$\nabla_\pi \mathcal{L}(y, \pi) = -A'y + b = 0 \quad (4.122b)$$

*Proof.* See Nocedal and Wright (1999).  $\square$

The first order necessary and sufficient optimality conditions (4.122) may be stated and solved as the KKT-system presented in the following corollary. In practice, the optimal solution of (4.120) is obtained by solving the symmetric indefinite system of equations constituting the KKT-system, i.e. (4.123).

**Corollary 4.4.4**

Let  $y \in \mathbb{R}^n$  and  $\pi \in \mathbb{R}^m$ . Then  $y$  is a global minimizer of (4.120) if and only if

$$\begin{pmatrix} G & -A \\ -A' & 0 \end{pmatrix} \begin{pmatrix} y \\ \pi \end{pmatrix} = - \begin{pmatrix} g \\ b \end{pmatrix} \quad (4.123)$$

*Proof.* Follows directly from proposition 4.4.3.  $\square$

**Remark 4.4.5**

The matrix

$$\begin{pmatrix} G & -A \\ -A' & 0 \end{pmatrix} \quad (4.124)$$

is called the Karush-Kuhn-Tucker (KKT) matrix of (4.120).

**Proposition 4.4.6**

Let the KKT-matrix of (4.120) be non-singular. Let  $\pi \in \mathbb{R}^m$  and  $y \in \mathbb{R}^n$ . Then  $x$  is the unique global minimizer of (4.120) if and only if

$$\begin{pmatrix} G & -A \\ -A' & 0 \end{pmatrix} \begin{pmatrix} y \\ \pi \end{pmatrix} = - \begin{pmatrix} g \\ b \end{pmatrix} \quad (4.125)$$

*Proof.* By corollary 4.4.4, it has been established that  $y$  is a global minimizer of (4.120) if and only if

$$\begin{pmatrix} G & -A \\ -A' & 0 \end{pmatrix} \begin{pmatrix} y \\ \pi \end{pmatrix} = - \begin{pmatrix} g \\ b \end{pmatrix} \quad (4.126)$$

As the KKT-matrix is non-singular,  $(y, \pi)$  determined by this KKT-system is unique. Consequently,  $y$  is the unique global minimizer of (4.120).  $\square$

If the matrix  $A \in \mathbb{R}^{n \times m}$  does not have full column rank, then either the constraints are too stringent such that no feasible solution exists or some of the constraints are superfluous. In both cases the KKT-matrix is singular. However, in the case of superfluous constraints, these can be discarded from the problem as they are linear combinations of other constraints. Therefore we may assume without loss of generality for feasible problems that the matrix  $A \in \mathbb{R}^{n \times m}$  has full column rank.



**Proposition 4.4.7**

Let  $A \in \mathbb{R}^{n \times m}$  have full column rank. Let  $Z \in \mathbb{R}^{n \times (n-m)}$  be the null space of  $A'$ . Let  $G \in \mathbb{R}^{n \times n}$  be symmetric positive semi-definite. Then the KKT-matrix

$$\begin{pmatrix} G & -A \\ -A' & 0 \end{pmatrix} \quad (4.127)$$

is non-singular if and only if  $Z'GZ$  is positive definite.

*Proof.* See Nocedal and Wright (1999).  $\square$

In summary, it has been established that if the KKT-matrix (4.124) of the equality constrained convex quadratic program (4.120) is non-singular, then the unique global minimizer,  $y$ , of (4.120) may be obtained as the solution of the KKT-system (4.123).

#### 4.4.1 Extended Linear-Quadratic Optimal Control Problem

In this subsection, we consider the extended linear-quadratic optimal control problem (4.46) as a convex quadratic program. The extended linear-quadratic optimal control problem is

$$\min_{\{x_{k+1}, u_k\}_{k=0}^{N-1}} \phi = \sum_{k=0}^{N-1} l_k(x_k, u_k) + l_N(x_N) \quad (4.128a)$$

$$s.t. \quad x_{k+1} = A'_k x_k + B'_k u_k + b_k \quad k = 0, 1, \dots, N-1 \quad (4.128b)$$

with the stage costs

$$l_k(x_k, u_k) = \frac{1}{2} x'_k Q_k x_k + x'_k M_k u_k + \frac{1}{2} u'_k R_k u_k + q'_k x_k + r'_k u_k + f_k \quad k = 0, 1, \dots, N-1 \quad (4.129a)$$

$$l_N(x_N) = \frac{1}{2} x'_N P_N x_N + p'_N x_N + \gamma_N \quad (4.129b)$$

In the extended linear-quadratic optimal control problem, the initial state  $x_0$  is a parameter and not a decision variable. To emphasize this fact, we may express the stage costs as

$$l_0(x_0, u_0) = \frac{1}{2} u'_0 R_0 u_0 + (M'_0 x_0 + r_0)' u_0 + \left( \frac{1}{2} x'_0 Q_0 x_0 + q'_0 x_0 + f_0 \right) \quad (4.130a)$$

$$l_k(x_k, u_k) = \frac{1}{2} \begin{pmatrix} x_k \\ u_k \end{pmatrix}' \begin{pmatrix} Q_k & M_k \\ M'_k & R_k \end{pmatrix} \begin{pmatrix} x_k \\ u_k \end{pmatrix} + \begin{pmatrix} q_k \\ r_k \end{pmatrix}' \begin{pmatrix} x_k \\ u_k \end{pmatrix} + f_k \quad k = 1, 2, \dots, N-1 \quad (4.130b)$$

$$l_N(x_N) = \frac{1}{2} x'_N P_N x_N + p'_N x_N + \gamma_N \quad (4.130c)$$

Similarly, the constraints may be expressed as

$$B'_0 u_0 - x_1 = -(A'_0 x_0 + b_0) \quad (4.131a)$$

$$A'_k x_k + B'_k u_k - x_{k+1} = -b_k \quad k = 1, 2, \dots, N-1 \quad (4.131b)$$

in which the parameters are on the right hand side of the equality sign.

To illustrate the formulation of the extended linear-quadratic optimal control problem as a quadratic program, consider the case with a horizon  $N = 3$ . In this case, the extended linear-quadratic optimal control problem may be formulated as an equality constrained quadratic program

$$\min_y \quad \phi = \frac{1}{2} y' G y + g' y + \rho \quad (4.132a)$$

$$s.t. \quad A' y = b \quad (4.132b)$$

with the vector of decision variables,  $y$ , given by

$$y = \begin{pmatrix} u_0 \\ x_1 \\ u_1 \\ x_2 \\ u_2 \\ x_3 \end{pmatrix} \quad (4.133)$$

The parameters of the objective function in (4.132) may be expressed as

$$G = \begin{pmatrix} R_0 & & & & & \\ & Q_1 & M_1 & & & \\ & M'_1 & R_1 & & & \\ & & & Q_2 & M_2 & \\ & & & M'_2 & R_2 & \\ & & & & & P_3 \end{pmatrix} \quad g = \begin{pmatrix} r_0 + M'_0 x_0 \\ q_1 \\ r_1 \\ q_2 \\ r_2 \\ p_3 \end{pmatrix} \quad (4.134a)$$

$$\rho = \left( \frac{1}{2} x'_0 Q_0 x_0 + q'_0 x_0 + f_0 \right) + \sum_{k=1}^2 f_k + \gamma_3 \quad (4.134b)$$

Similarly, the parameters defining the equality constraints are given by the expressions

$$A = \begin{pmatrix} B_0 & & & \\ -I & A_1 & & \\ & B_1 & & \\ & -I & A_2 & \\ & & B_2 & \\ & & & -I \end{pmatrix} \quad b = - \begin{pmatrix} A'_0 x_0 + b_0 \\ b_1 \\ b_2 \end{pmatrix} \quad (4.135)$$

As the identity matrix,  $I$ , has full rank it is trivial to establish that the matrix  $A$  has full column rank.

With the just introduced expressions for  $G$  and  $A$ , it is clear that the KKT-matrix of the extended linear-quadratic optimal control problem (4.46) may be expressed as

$$\begin{pmatrix} G & -A \\ -A' & 0 \end{pmatrix} = \left( \begin{array}{cccc|ccc} R_0 & & & & -B_0 & & & \\ & Q_1 & M_1 & & I & -A_1 & & \\ & M_1' & R_1 & & & -B_1 & & \\ & & & Q_2 & M_2 & & -A_2 & \\ & & & M_2' & R_2 & & -B_2 & \\ & & & & & P_3 & & I \end{array} \right) \quad (4.136)$$

If assumption 4.3.3 is satisfied, then  $G$  of the extended linear-quadratic optimal control problem is symmetric positive semi-definite. Furthermore, if  $R_k$  is positive definite for  $k = 0, 1, \dots, N-1$ , i.e. assumption 4.3.4 is satisfied, then the KKT-matrix of the extended linear-quadratic optimal control problem is non-singular. This fact is readily established by observing the KKT-matrix. Each block column involving a matrix  $R_k$  has full column rank due to assumption 4.3.4. All other block columns do also have full column rank as they involve the identity matrix,  $I$ . Furthermore, due to the structure of the KKT-matrix all block columns are linearly independent. Therefore, the KKT-matrix has full column rank and is non-singular as it is a square matrix.

**Lemma 4.4.8**

Let assumptions 4.3.3 and 4.3.4 be satisfied. Then the KKT-matrix of the extended linear-quadratic optimal control problem (4.46) is non-singular.

*Proof.* Follows from the preceding discussion.  $\square$

**Remark 4.4.9**

Positive definiteness of  $R_k$  for  $k = 0, 1, \dots, N-1$  (i.e. assumption 4.3.4) is a sufficient but not a necessary condition for non-singularity of the KKT-matrix. Using the argument from the dynamic programming approach a necessary and sufficient condition for non-singularity of the KKT-matrix is that the matrices

$$R_{e,k} = R_k + B_k P_{k+1} B_k' \quad k = 0, 1, \dots, N-1 \quad (4.137)$$

are non-singular.

In the following proposition we state necessary and sufficient conditions for the characterization of a unique global minimizer of the extended linear-quadratic optimal control problem. These conditions are based on the necessary and sufficient conditions characterizing the unique global minimizer of a convex quadratic program.

**Proposition 4.4.10**

Let assumptions 4.3.3 and 4.3.4 be satisfied. Let  $\pi_k \in \mathbb{R}^n$  for  $k = 0, 1, \dots, N-1$ . Let  $x_0 \in \mathbb{R}^n$  be given.

Then  $\{x_{k+1}, u_k\}_{k=0}^{N-1}$  is the unique global minimizer of the extended linear-quadratic optimal control problem (4.46) if and only if

$$M'_k x_k + R_k u_k + r_k - B_k \pi_k = 0 \quad k = 0, 1, \dots, N-1 \quad (4.138a)$$

$$Q_k x_k + M_k u_k + q_k - A_k \pi_k + \pi_{k-1} = 0 \quad k = 1, 2, \dots, N-1 \quad (4.138b)$$

$$P_N x_N + p_N + \pi_{N-1} = 0 \quad (4.138c)$$

$$A'_k x_k + B'_k u_k + b_k - x_{k+1} = 0 \quad k = 0, 1, \dots, N-1 \quad (4.138d)$$

*Proof.* The Lagrangian of the extended linear-quadratic optimal control problem (4.46) is

$$\begin{aligned} \mathcal{L}(\{x_{k+1}, u_k, \pi_k\}_{k=0}^{N-1}) &= \sum_{k=0}^{N-1} l_k(x_k, u_k) + l_N(x_N) - \sum_{k=0}^{N-1} \pi'_k (A'_k x_k + B'_k u_k + b_k - x_{k+1}) \\ &= \sum_{k=0}^{N-1} \frac{1}{2} \begin{pmatrix} x_k \\ u_k \end{pmatrix}' \begin{pmatrix} Q_k & M_k \\ M'_k & R_k \end{pmatrix} \begin{pmatrix} x_k \\ u_k \end{pmatrix} + \begin{pmatrix} q_k \\ r_k \end{pmatrix}' \begin{pmatrix} x_k \\ u_k \end{pmatrix} + f_k \\ &\quad + \frac{1}{2} x'_N P_N x_N + p'_N x_N + \gamma_N \\ &\quad - \sum_{k=0}^{N-1} \pi'_k (A'_k x_k + B'_k u_k + b_k - x_{k+1}) \\ &= \sum_{k=0}^{N-1} \frac{1}{2} (x'_k Q_k x_k + 2x'_k M_k u_k + u'_k R_k u_k) + q'_k x_k + r'_k u_k + f_k \\ &\quad + \frac{1}{2} x'_N P_N x_N + p'_N x_N + \gamma_N \\ &\quad - \sum_{k=0}^{N-1} \pi'_k (A'_k x_k + B'_k u_k + b_k - x_{k+1}) \end{aligned} \quad (4.139)$$

The first-order necessary and sufficient optimality conditions are

$$\nabla_{u_k} \mathcal{L} = M'_k x_k + R_k u_k + r_k - B_k \pi_k = 0 \quad k = 0, 1, \dots, N-1$$

(4.140a)

$$\nabla_{x_k} \mathcal{L} = Q_k x_k + M_k u_k + q_k - A_k \pi_k + \pi_{k-1} = 0 \quad k = 1, 2, \dots, N-1$$

(4.140b)

$$\nabla_{x_N} \mathcal{L} = P_N x_N + p_N + \pi_{N-1} = 0$$

(4.140c)

$$\nabla_{\pi_k} \mathcal{L} = -(A'_k x_k + B'_k u_k + b_k - x_{k+1}) = 0 \quad k = 0, 1, \dots, N-1$$

(4.140d)

Consequently, as the associated KKT-matrix is non-singular according to lemma 4.4.8 the unique global minimizer of the extended linear-quadratic optimal control problem, which is a quadratic program, is characterized by the necessary and sufficient conditions

$$M'_k x_k + R_k u_k + r_k - B_k \pi_k = 0 \quad k = 0, 1, \dots, N-1 \quad (4.141a)$$

$$Q_k x_k + M_k u_k + q_k - A_k \pi_k + \pi_{k-1} = 0 \quad k = 1, 2, \dots, N-1 \quad (4.141b)$$

$$P_N x_N + p_N + \pi_{N-1} = 0 \quad (4.141c)$$

$$A'_k x_k + B'_k u_k + b_k - x_{k+1} = 0 \quad k = 0, 1, \dots, N-1 \quad (4.141d)$$

□

For illustrative and notational purposes let the prediction horizon be  $N = 3$ . For the general convex equality constrained quadratic program, the first order necessary and sufficient conditions may be stated as the KKT-system

$$\begin{pmatrix} G & -A \\ -A' & 0 \end{pmatrix} \begin{pmatrix} y \\ \pi \end{pmatrix} = - \begin{pmatrix} g \\ b \end{pmatrix} \quad (4.142)$$

Under the appropriate assumptions discussed previously, the KKT-matrix is non-singular and the unique global minimizer of (4.120) may be obtained as part of the solution of the above KKT-system. Similarly, the first order necessary and sufficient optimality conditions for the extended linear-quadratic optimal control problem may be stated as the following KKT-system for the

case  $N = 3$

$$\left( \begin{array}{cccc|ccc} R_0 & & & & -B_0 & & \\ & Q_1 & M_1 & & I & -A_1 & \\ & M'_1 & R_1 & & & -B_1 & \\ & & & Q_2 & M_2 & & \\ & & & M'_2 & R_2 & & \\ & & & & & I & -A_2 \\ & & & & & & -B_2 \\ & & & & & & I \\ \hline -B'_0 & I & & & & & \\ & -A'_1 & -B'_1 & I & & & \\ & & & -A'_2 & -B'_2 & I & \end{array} \right) \begin{pmatrix} u_0 \\ x_1 \\ u_1 \\ x_2 \\ u_2 \\ x_3 \\ \pi_0 \\ \pi_1 \\ \pi_2 \end{pmatrix} = - \begin{pmatrix} M'_0 x_0 + r_0 \\ q_1 \\ r_1 \\ q_2 \\ r_2 \\ p_3 \\ -A'_0 x_0 - b_0 \\ -b_1 \\ -b_2 \end{pmatrix} \quad (4.143)$$

Under assumptions 4.3.3 and 4.2.4, the unique global minimizer,  $\{x_{k+1}, u_k\}_{k=0}^{N-1=2}$ , of the corresponding extended linear-quadratic optimal control problem may be computed as part of the solution of the above highly structured KKT-system. By appropriate reordering of the decision variables and Lagrange multipliers, we obtain a banded KKT-system which is equivalent to (4.143)

$$\left( \begin{array}{cccccccc|cccc} R_0 & -B_0 & & & & & & & & & \\ -B'_0 & 0 & I & & & & & & & & \\ & I & Q_1 & M_1 & -A_1 & & & & & & \\ & & M'_1 & R_1 & -B_1 & & & & & & \\ & & -A'_1 & -B'_1 & 0 & I & & & & & \\ & & & & I & Q_2 & M_2 & -A_2 & & & \\ & & & & & M'_2 & R_2 & -B_2 & & & \\ & & & & & -A'_2 & -B'_2 & 0 & I & & \\ & & & & & & & I & P_3 & & \end{array} \right) \begin{pmatrix} u_0 \\ \pi_0 \\ x_1 \\ u_1 \\ \pi_1 \\ x_2 \\ u_2 \\ \pi_2 \\ x_3 \end{pmatrix} = - \begin{pmatrix} M'_0 x_0 + r_0 \\ -A'_0 x_0 - b_0 \\ q_1 \\ r_1 \\ -b_1 \\ q_2 \\ r_2 \\ -b_2 \\ p_3 \end{pmatrix} \quad (4.144)$$

The solution of this banded system of linear equations may be accomplished using a solver for banded systems (c.f. Golub and Van Loan, 1996). DGBSV in LAPACK (c.f. Anderson *et al.*, 1999) is a solver for band diagonal systems. The method employed in DGBSV is banded LU-factorization with partial pivoting (see also Wright, 1996).

The unique solution,  $\{x_{k+1}, u_k, \pi_k\}_{k=0}^{N-1}$ , of (4.143) and (4.144) consists of the unique global minimizer,  $\{x_{k+1}, u_k\}_{k=0}^{N-1}$ , of the extended linear-quadratic optimal control problem (4.46) and of its associated Lagrange multipliers  $\{\pi_k\}_{k=0}^{N-1}$ . The unique global minimizer,  $\{x_{k+1}, u_k\}_{k=0}^{N-1}$ , of the extended linear-quadratic optimal control problem (4.46) may be obtained by dynamic programming as specified by proposition 4.3.5 and implemented by algorithms 1-3. Consequently, the solution of (4.143) and (4.144) may be obtained by applying proposition 4.3.5 for obtaining the primal solution  $\{x_{k+1}, u_k\}_{k=0}^{N-1}$  and subsequent application of the recursion

$$\pi_{N-1} = -P_N x_N - p_N \quad (4.145a)$$

$$\pi_{k-1} = A_k \pi_k - Q_k x_k - M_k u_k - q_k \quad k = N-1, N-2, \dots, 1 \quad (4.145b)$$

for obtaining the dual solution  $\{\pi_k\}_{k=0}^{N-1}$ . This is an efficient procedure for solution of the block banded system (4.144) as it is linear in the horizon,  $N$ , and scales cubically with the size of the blocks,  $A_k$  and  $B_k$ . This Riccati procedure explicitly utilizes the fact that the matrices connecting the blocks are identity matrices.

Algorithm 4 summarizes the procedure for computation of the solution of the extended linear-quadratic optimal control problem and its associated Lagrange multipliers. The solution,  $\{x_{k+1}, u_k, \pi_k\}_{k=0}^{N-1}$ , provided corresponds to the solution when the extended linear-quadratic optimal control problem is regarded as a quadratic program. Algorithm 4 may also be regarded as a Riccati based method for factorization and solution of either (4.143) or (4.144). In this algorithm, the factorization of the KKT-matrix is conducted simultaneous with the solution of the KKT-system. This implies that the sequence of matrices,  $P_k$ , are not stored.

Algorithm 5 is algorithm 3 extended with computation of the Lagrange multipliers  $\{\pi_k\}_{k=0}^{N-1}$  according to (4.145). Application of this algorithm requires that a factorization  $\{P_k, L_k, Z_k\}_{k=0}^{N-1}$  of the KKT-matrix corresponding to the extended linear-quadratic optimal control problem has already been computed. This factorization may be conducted using algorithm 2. Algorithm 5 is supposed to be used when the extended linear-quadratic optimal control problem with constant  $(\{Q_k, M_k, R_k, A_k, B_k\}, P_N)$  is solved several times for different values of  $(x_0, \{q_k, r_k, b_k\}_{k=0}^{N-1}, p_N)$ . Consequently,  $\{P_k, L_k, Z_k\}_{k=0}^{N-1}$  in addition with the data  $(\{Q_k, M_k, R_k, A_k, B_k\}_{k=0}^{N-1}, P_N)$  may be regarded as a factorization of the KKT matrix in (4.143) as well as (4.144). Algorithm 2 is constructed for computation of this factorization. Algorithm 5 may be regarded as a procedure for solving (4.143) and (4.144) when their factorization  $\{P_k, L_k, Z_k\}_{k=0}^{N-1}$  is available.

Occasionally, it is more convenient to use the Lagrange multipliers,  $\mu_k$ , defined as  $\mu_k = -\pi_k$  rather than  $\pi_k$ . In this case, the KKT system corresponding to (4.143) may be expressed as

$$\left( \begin{array}{cccc|cc} R_0 & & & & B_0 & \\ & Q_1 & M_1 & & -I & A_1 \\ & M'_1 & R_1 & & & B_1 \\ & & & Q_2 & & A_2 \\ & & & M'_2 & & B_2 \\ & & & & P_3 & -I \end{array} \right) \begin{pmatrix} u_0 \\ x_1 \\ u_1 \\ x_2 \\ u_2 \\ x_3 \\ \mu_0 \\ \mu_1 \\ \mu_2 \end{pmatrix} = - \begin{pmatrix} M'_0 x_0 + r_0 \\ q_1 \\ r_1 \\ q_2 \\ r_2 \\ p_3 \\ A'_0 x_0 + b_0 \\ b_1 \\ b_2 \end{pmatrix} \quad (4.165)$$

---

**Algorithm 4** Solution of the extended linear-quadratic optimal control problem as a QP.

---

**Require:**  $N$ ,  $(P_N, p_N, \gamma_N)$ ,  $\{Q_k, M_k, R_k, q_k, f_k, r_k, A_k, B_k, b_k\}_{k=0}^{N-1}$ , and  $x_0$ .

Assign  $P \leftarrow P_N$ ,  $p \leftarrow p_N$ , and  $\gamma \leftarrow \gamma_N$ .

**for**  $k = N - 1 : -1 : 0$  **do**

    Compute the temporary matrices and vectors

$$R_e = R_k + B_k P B_k' \quad (4.146a)$$

$$S = A_k P \quad (4.146b)$$

$$Y = (M_k + S B_k')' \quad (4.146c)$$

$$s = P b_k \quad (4.146d)$$

$$c = s + p \quad (4.146e)$$

$$d = r_k + B_k c \quad (4.146f)$$

    Cholesky factorize  $R_e$

$$R_e = L_k L_k' \quad (4.147)$$

    Compute  $Z_k$  and  $z_k$  by solving

$$L_k Z_k = Y \quad (4.148a)$$

$$L_k z_k = d \quad (4.148b)$$

    Update  $P$ ,  $\gamma$ , and  $p$  by

$$P \leftarrow Q_k + S A_k' - Z_k' Z_k \quad (4.149a)$$

$$\gamma \leftarrow \gamma + f_k + p' b_k + \frac{1}{2} s' b_k - \frac{1}{2} z_k' z_k \quad (4.149b)$$

$$p \leftarrow q_k + A_k c - Z_k' z_k \quad (4.149c)$$

**end for**

Compute the optimal value by

$$\phi = \frac{1}{2} x_0' P x_0 + p' x_0 + \gamma \quad (4.150)$$

**for**  $k = 0 : 1 : N - 1$  **do**

    Compute

$$y = Z_k x_k + z_k \quad (4.151)$$

    and solve the linear system of equations

$$L_k' u_k = -y \quad (4.152)$$

    for  $u_k$ .

    Compute

$$x_{k+1} = A_k' x_k + B_k' u_k + b_k \quad (4.153)$$

**end for**

Compute

$$\pi_{N-1} = -P_{N-1} x_{N-1} - p_{N-1} \quad (4.154)$$

**for**  $k = N - 1 : -1 : 1$  **do**

    Compute

$$\pi_{k-1} = A_k \pi_k - Q_k x_k - M_k u_k - q_k \quad (4.155)$$

**end for**

Return  $\{x_{k+1}, u_k, \pi_k\}_{k=0}^{N-1}$  and  $\phi$ .

---



---

**Algorithm 5** Solve a factorized extended linear-quadratic optimal control problem as a QP.

---

**Require:**  $N$ ,  $(P_N, p_N, \gamma_N)$ ,  $\{Q_k, M_k, R_k, q_k, r_k, f_k, A_k, B_k, b_k\}_{k=0}^{N-1}$ ,  $x_0$ , and  $\{P_k, L_k, Z_k\}_{k=0}^{N-1}$ .

Assign  $p \leftarrow p_N$  and  $\gamma \leftarrow \gamma_N$ .

**for**  $k = N - 1 : -1 : 0$  **do**

    Compute the temporary vectors

$$s = P_{k+1}b_k \quad (4.156a)$$

$$c = s + p \quad (4.156b)$$

$$d = r_k + B_k c \quad (4.156c)$$

    Solve the lower triangular system of equations

$$L_k z_k = d \quad (4.157)$$

    for  $z_k$ .

    Update  $\gamma$  and  $p$  by the expressions

$$\gamma \leftarrow \gamma + f_k + p' b_k + \frac{1}{2} s' b_k - \frac{1}{2} z_k' z_k \quad (4.158a)$$

$$p \leftarrow q_k + A_k c - Z_k' z_k \quad (4.158b)$$

**end for**

Compute the optimal value

$$\phi = \frac{1}{2} x_0' P_0 x_0 + p' x_0 + \gamma \quad (4.159)$$

**for**  $k = 0 : 1 : N - 1$  **do**

    Compute

$$y = Z_k x_k + z_k \quad (4.160)$$

    and solve the upper triangular system of equations

$$L_k' u_k = -y \quad (4.161)$$

    for  $u_k$ .

    Compute

$$x_{k+1} = A_k' x_k + B_k' u_k + b_k \quad (4.162)$$

**end for**

Compute

$$\pi_{N-1} = -P_{N-1} x_{N-1} - p_{N-1} \quad (4.163)$$

**for**  $k = N - 1 : -1 : 1$  **do**

    Compute

$$\pi_{k-1} = A_k \pi_k - Q_k x_k - M_k u_k - q_k \quad (4.164)$$

**end for**

Return  $\{x_{k+1}, u_k, \pi_k\}_{k=0}^{N-1}$  and  $\phi$ .

---

and the KKT-system corresponding to (4.144) may be expressed as

$$\begin{pmatrix} R_0 & B_0 & & & & & & & \\ B'_0 & 0 & -I & & & & & & \\ & -I & Q_1 & M_1 & A_1 & & & & \\ & & M'_1 & R_1 & B_1 & & & & \\ & & A'_1 & B'_1 & 0 & -I & & & \\ & & & -I & Q_2 & M_2 & A_2 & & \\ & & & & M'_2 & R_2 & B_2 & & \\ & & & & A'_2 & B'_2 & 0 & -I & \\ & & & & & & -I & P_3 & \end{pmatrix} \begin{pmatrix} u_0 \\ \mu_0 \\ x_1 \\ u_1 \\ \mu_1 \\ x_2 \\ u_2 \\ \mu_2 \\ x_3 \end{pmatrix} = - \begin{pmatrix} M'_0 x_0 + r_0 \\ A'_0 x_0 + b_0 \\ q_1 \\ r_1 \\ b_1 \\ q_2 \\ r_2 \\ b_2 \\ p_3 \end{pmatrix} \quad (4.166)$$

Either (4.165) or (4.166) may be solved by computing  $\{x_{k+1}, u_k\}_{k=0}^{N-1}$  using the Riccati based iteration specified in proposition 4.3.5 and subsequent computation of  $\mu_k$  by

$$\mu_{N-1} = P_N x_N + p_N \quad (4.167a)$$

$$\mu_{k-1} = A_k \mu_k + Q_k x_k + M_k u_k + q_k \quad k = N-1, N-2, \dots, 1 \quad (4.167b)$$

## 4.5 Conclusion

The extended linear-quadratic optimal control problem (4.46) has been defined as a finite horizon linear-quadratic program with control structure. In particular, the stage costs of the extended linear-quadratic optimal control problem contain terms linear in the states,  $x_k$ , and terms linear in the controls,  $u_k$ . Furthermore, the dynamic equations contain an affine term,  $b_k$ . These terms are the extensions of the extended linear-quadratic optimal control problem compared to the standard linear-quadratic optimal control problem (4.15) usually considered in the engineering control literature.

Using dynamic programming, a computational efficient Riccati based procedure for solution of the extended linear-quadratic optimal control problem has been developed. The developed Riccati based procedure may also be used to factorize and solve the KKT-system corresponding to the extended linear-quadratic optimal control problem.

The extended linear-quadratic optimal control problem constitute the quadratic subproblem in SQP procedures for nonlinear optimal control. Furthermore, the linear terms in the objective function and the affine term in the dynamic equations are important in linear model predictive control with infeasibilities and anticipatory linear model predictive control.



# Numerical Solution of Nonlinear Optimal Control Problems

In a tutorial fashion, the principles for efficient solution of unconstrained nonlinear optimal control problems described by ordinary differential equations are presented. These principles are presented through numerical solution of a continuous-time nonlinear optimal control problem of the Bolza form. To focus on the basic principles involved and for illustrative purposes, the continuous-time Bolza problem is discretized by the explicit Euler method. The discrete-time nonlinear optimal control problem of the Bolza form is solved by different SQP methods and an algorithm based on the discrete maximum principle. The SQP algorithms presented are implementations based on open- and closed-loop feasible path control vector parameterizations as well as an infeasible path simultaneous procedure. Two procedures for solution of the quadratic programs arising in the solution of the nonlinear optimal control problem are presented. In the first procedure, the structure of the quadratic programs arising in the solution of the nonlinear optimal control problem is utilized by a Riccati iteration based factorization of the resulting KKT-system. In the second procedure, an efficient procedure for elimination of the states and solution of a dense reduced space quadratic program is presented. These methods are compared for a simple process example operated around an unstable equilibrium. The infeasible path and the closed-loop feasible path algorithms converge for this example. The implemented open-loop feasible path algorithms are not able to converge to an unstable equilibrium. The Riccati based solution procedure enables implementation of the stabilized infeasible path SQP algorithm as well as the closed-loop feasible path SQP algorithm. The methods are presented in a framework that is easily extended to constrained nonlinear optimal control problems. Such extensions and methodologies for efficient integration of the ordinary differential equations as well as the corresponding sensitivity equations are discussed.

## 5.1 Introduction

Process optimization, and dynamic optimization in particular, is one of the fastest growing technologies in the automation industry because of its ability to link a company's business and economic objectives to its operations. Dynamic optimization is the only technology that can truly optimize state transitions in continuous processes and it is equally applicable to batch and semi-batch processes. Therefore, dynamic process optimization is one of the most efficient ways to achieve optimal asset utilization and performance, as it is often aimed directly at improving plant profitability in an immediately quantifiable way. Almost all processes can benefit from dynamic optimization in some way - by increasing yield and throughput, limiting off-spec production, reducing downtime, and lowering energy costs.

Dynamic optimization concerns the solution of optimization problems constrained by systems of differential equations. This problem is also known as the deterministic nonlinear optimal control problem, or just the nonlinear optimal control problem. Numerical solution of the nonlinear optimal control problem is the key enabling technique for moving horizon estimation and control. Collectively, moving horizon estimation and control are known as model predictive control, which has received wide spread industrial acceptance as the preferred advanced process control technique.

In this paper specialized algorithms for numerical solution of the unconstrained nonlinear optimal control problem are presented. We advocate the use of specialized algorithms rather than bundling of off-the-shelf algorithms for optimization and integration. By the specialized approach solution of large-scale systems as well as unstable systems is made possible. Furthermore, by the development of specialized algorithms, inherent numerical problems associated with control of systems around unstable equilibria as well as efficient linear algebra techniques for solution of the linear system arising are revealed. These insights extends and refines the theory of linear systems and the optimal control of linear quadratic systems.

This paper is limited to unconstrained optimal control problems. That is optimal control problems in which the only constraint is the differential equations describing the dynamics of the system. Furthermore, the dynamics is restricted to be described by systems of ordinary differential equations. This restriction excludes systems whose dynamics is described by differential-algebraic equations. Finally, the methods are presented by fixed step size explicit integration of the system of ordinary differential equations. All these simplifications are introduced to direct the focus on the essential principles for numerical solution of optimal control problems. The methods presented can be extended to systems governed by index-1 differential-algebraic equations, integrated by an implicit integration method, and with constraints on the control inputs. The infeasible path sequential quadratic programming (SQP) algorithm can even be extended to systems with state constraints. However, the feasible path SQP

methods and the discrete maximum principle algorithm cannot in a conceptually simple fashion be extended to systems with state constraints such that feasibility with respect to the inequality state constraints is respected at every iteration of the algorithm. However, if violation of the state constraint during the solution process is acceptable, then the feasible path methods can be extended to systems with state constraints in a straightforward manner.

Utilization of the special structure arising from the separable objective function and the fact that the constraints are differential equations is essential for effective and practical solution of unconstrained optimal control problems. The structure exploiting algorithms are compared to algorithms constructing a dense quadratic program to be solved at each iteration. Numerical issues associated with dynamic optimization around unstable equilibria and the implications for the considered algorithms are discussed extensively.

The basic principles of sequential quadratic programming are presented in section 5.2. Section 5.3 provides an overview of numerical methods for solution of the continuous time optimal control problem. Furthermore, the principles of sequential quadratic programming are specialized to the unconstrained optimal control problem. The specialization consists of a partitioned update of the Hessian approximation and a structure utilizing Riccati based factorization of the KKT-matrix. The proposed sequential quadratic programming algorithm is an infeasible path method using an explicit Euler method with fixed step size for integration. Both an open-loop and a closed-loop feasible path SQP algorithm are presented in section 5.4. These algorithms are also based on factorization of the KKT-matrix by Riccati iteration. Section 5.6 propose an algorithm based on the discrete maximum-principle rather than sequential quadratic programming. In section 5.7, the algorithms are demonstrated on a process example concerning operation at an unstable equilibrium. Extensions and refinements of the algorithms are presented and discussed in section 5.8.

## 5.2 Sequential Quadratic Programming

In this section, we review the optimality conditions for an equality constrained nonlinear program. These conditions are used to establish a sequential quadratic programming algorithm based on line search for the equality constrained nonlinear program. The developed algorithm is intended to serve as a template for unconstrained nonlinear optimal control algorithms and make the construction of such algorithms transparent.

Consider the equality constrained nonlinear program

$$\min_{x \in \mathbb{R}^n} f(x) \tag{5.1a}$$

$$s.t. \quad g(x) = 0 \tag{5.1b}$$

in which  $f : \mathbb{R}^n \mapsto \mathbb{R}$  and  $g : \mathbb{R}^n \mapsto \mathbb{R}^m$  are twice continuously differentiable

functions. Sequential quadratic programming solves the nonlinear program (5.1) by solution of a sequence of equality constrained quadratic programs.

The sequential quadratic programming algorithm was first developed by Wilson (1963) who used the exact Hessian of the Lagrangian function. Garcia-Palomares and Mangasarian (1976) first suggested to use variable-metric methods for approximation of the quadratic part in the quadratic programs solved by the sequential quadratic programming algorithm. Han (1976) showed that if the Hessian in the QP problem is replaced by an approximation computed by either the Davidson-Fletcher-Powell update (DFP) or Powell's symmetric Broyden update (PSB), then the resulting SQP method is locally superlinearly convergent in both the primal and the dual variables. Furthermore, locally it takes constant step sizes ( $\alpha = 1$ ). In developing these results, Han required that the KKT-matrix is non-singular and that the initial Hessian approximation is sufficiently close to the true Hessian of the solution. Later, Han (1977) proved global convergence using a  $l_1$ -penalty function and assuming that the Lagrange multipliers remain bounded at all iterations. From a practical point of view, Han's results leaves many open problems. Some unresolved issues concern selection of the penalty parameter in the  $l_1$ -penalty function and how to guarantee boundedness of the approximate Hessian for a sequence of quasi-Newton updates. Han did not present any numerical results. Powell (1977) was the first to present a working SQP method based on a variable-metric approximation of the Hessian matrix. Powell suggested a modified Broyden-Fletcher-Goldfarb-Shanno (BFGS) update which keeps the Hessian approximations positive definite. On the basis of empirical testing, Powell proposed a modified  $l_1$ -penalty function for the line search and gave a procedure for computation of the penalty weights. Much of the practical success of Powell's algorithm depends on heuristic elements. Due to these heuristics a number of difficulties have been occasionally observed:

1. *Marattos effect* (c.f. Marattos, 1978). It may happen that a full step is not taken even though the solution is close to the optimal solution and the approximate Hessian approximates the true Hessian well. This implies that the rate of convergence is only linear. Chamberlain *et al.* (1982), Chen and Stadtherr (1984), and Hoza and Stadtherr (1993) have proposed to use the watchdog technique to overcome this difficulty.
2. *Cycling*. In some examples Powell's SQP algorithm cycles (c.f. Chamberlain, 1979). This is attributed to the selection of weights in Powell's  $l_1$ -penalty function. The watchdog technique can also be used to overcome this problem.
3. *Unboundedness of the Hessian approximation*. Chamberlain (1979) shows in a simple example that the approximate Hessian matrix may not remain bounded when the number of iterations become large.
4. *Ill-conditioning of the Hessian approximation*. The modified BFGS up-

date can give highly ill-conditioned Hessian approximations, which invalidates the computed search directions (c.f. Powell, 1985b).

5. *Inconsistent linearized constraints.* For large-scale problems the linearization of the constraints is not necessarily consistent, which may leave the resulting QP infeasible. Biegler and Cuthrell (1985) have proposed constraint relaxation procedures to avoid the problems with inconsistent constraints due to the linearization. However, inconsistent linearized constraints are not possible in unconstrained optimal control problems.

Despite these problems, SQP methods perform well in practice and are considered as some of the most efficient methods for solution of nonlinearly constrained optimization problems. The primary advantage of SQP methods over other gradient-based methods is the low number of function evaluations needed to converge to a solution. This is a particularly attractive property, when function evaluations are expensive as in the optimal control problem.

Boggs and Tolle (1996, 2000) provide recent overviews of large-scale sequential quadratic programming. Gill *et al.* (1997) describe the algorithmic ideas implemented in the popular large-scale SQP algorithm, SNOPT.

### 5.2.1 Optimality Conditions

The Lagrangian function is central in the development of a solution procedure for constrained nonlinear optimization problems. The Lagrangian function of (5.1) is

$$\mathcal{L}(x, \pi) = f(x) - \pi' g(x) \quad (5.2)$$

To be able to state optimality conditions for nonlinear programs, some kind of constraint qualification is necessary (c.f. Mangasarian, 1994). The constraint qualification typically invoked is the linear independence constraint qualification (LICQ).

**Definition 5.2.1 (LICQ)**

Given a point  $x$ . The linear independence constraint qualification (LICQ) holds if  $\nabla g(x)$  has full column rank.

The linear independence constraint qualification is not a necessary condition for characterization of a minimizer of (5.1). However, it is a sufficient condition and very convenient. As explained by ?) the linear independence constraint qualification guarantees that a direction  $p$  satisfying  $\nabla g(x^0)'p = 0$  at any feasible point  $x^0$  is the tangent of a feasible arc emanating from the current point  $x^0$ . Therefore the linear independence constraint qualification satisfies the first-order constraint qualification which is necessary in the proof of Kuhn-Tucker optimality conditions. Essentially, the linear constraint qualification can be regarded as a condition that ensures that the linear approximation to the feasible region at  $x^*$  captures the essential geometric features of the true feasible set in



some neighborhood of  $x^*$ . In particular, it should be noted that no gradient  $\nabla g_i(x^*)$  can be zero at the optimal solution under the linear independence constraint qualification. Further, the linear independence constraint qualification has been adopted because it is a necessary condition for non-singularity of the KKT-matrix used in computing search directions.

With the linear independence constraint qualification, the famous first order Karush-Kuhn-Tucker conditions may be stated as in the following proposition.

**Proposition 5.2.2 (Necessary KKT-Conditions)**

Let  $x^* \in \mathbb{R}^n$  be a local minimizer of (5.1) and let LICQ hold at  $x^*$ . Then there exists a unique Lagrange multiplier vector  $\pi^* \in \mathbb{R}^m$  such that

$$\nabla_x \mathcal{L}(x^*, \pi^*) = \nabla f(x^*) - \nabla g(x^*)\pi^* = 0 \quad (5.3a)$$

$$\nabla_\pi \mathcal{L}(x^*, \pi^*) = -g(x^*) = 0 \quad (5.3b)$$

*Proof.* See Bertsekas (1995b) or Nocedal and Wright (1999) □

A point  $(x, \pi)$  that satisfies (5.3) is called a KKT-point. For continuously differentiable functions, the necessary first-order KKT-conditions (5.3) are not sufficient for  $x$  being a local minimizer of (5.1). To be sufficient conditions, the nonlinear program (5.1) must be convex as stated in the following proposition.

**Proposition 5.2.3 (Sufficient KKT-Conditions)**

Let  $f : \mathbb{R}^n \mapsto \mathbb{R}$  be convex and let  $g_i : \mathbb{R}^n \mapsto \mathbb{R}$  be linear for  $i = 1, 2, \dots, m$ . If  $(x^*, \pi^*)$  satisfies (5.3), then  $x^*$  is the global minimizer of (5.1).

*Proof.* See Bertsekas (1995b) □

Sufficient conditions for a local minimizer of a nonlinear program (5.1) with a general possibly non-convex objective function and general equality constraints need second order derivative information of the objective function,  $f : \mathbb{R}^n \mapsto \mathbb{R}$ , and the constraint functions,  $g : \mathbb{R}^n \mapsto \mathbb{R}^m$ . The second order necessary condition for a local minimizer of (5.1) may be stated as in the following proposition.

**Proposition 5.2.4 (Second Order Necessary Condition)**

Let  $x^* \in \mathbb{R}^n$  be a local minimizer of (5.1) and let the LICQ condition be satisfied at  $x^*$ . Let  $\pi^* \in \mathbb{R}^m$  be a Lagrange vector such that (5.3) are satisfied. Then

$$p' \nabla_{xx}^2 \mathcal{L}(x^*, \pi^*) p \geq 0 \quad \forall p \in V(x^*) \quad (5.4)$$

in which

$$V(x^*) = \{p \in \mathbb{R}^n : \nabla g(x^*)' p = 0\} \quad (5.5)$$

*Proof.* See Bertsekas (1995b) □

To be sufficient for a local minimizer, the second order condition must be strengthened as in the proposition below.

**Proposition 5.2.5 (Second Order Sufficiency Condition)**

Let  $x^* \in \mathbb{R}^n$  and  $\pi^* \in \mathbb{R}^m$  satisfy (5.3) and

$$p' \nabla_{xx}^2 \mathcal{L}(x^*, \pi^*) p > 0 \quad \forall p \in V(x^*) \setminus \{0\} \quad (5.6)$$

Then  $x^*$  is a strict local minimizer of (5.1).

*Proof.* See Bertsekas (1995b) □

Hence, most algorithms locate a local minimizer by solving the KKT-conditions (5.3) and assume that (5.6) is satisfied as well. The KKT-conditions (5.3) are typically solved by a Newton or a quasi-Newton method. Solving (5.3) is equivalent to solving the system of nonlinear equations

$$F(x, \pi) = \begin{bmatrix} \nabla_x \mathcal{L}(x, \pi) \\ -g(x) \end{bmatrix} = \begin{bmatrix} \nabla f(x) - \nabla g(x) \pi \\ -g(x) \end{bmatrix} = 0 \quad (5.7)$$

By Newton's method, a search direction  $(\Delta x, \Delta \pi)$  from a current point  $(x^0, \pi^0)$  is computed by solving

$$\nabla_x F(x^0, \pi^0)' \Delta x + \nabla_\pi F(x^0, \pi^0)' \Delta \pi = -F(x^0, \pi^0) \quad (5.8)$$

which may be also be formulated as

$$\begin{bmatrix} \nabla_{xx}^2 \mathcal{L}(x^0, \pi^0) & -\nabla g(x^0) \\ -\nabla g(x^0)' & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta \pi \end{bmatrix} = - \begin{bmatrix} \nabla_x \mathcal{L}(x^0, \pi^0) \\ -g(x^0) \end{bmatrix} \quad (5.9)$$

in which the Hessian of the Lagrangian is

$$\nabla_{xx}^2 \mathcal{L}(x^0, \pi^0) = \nabla^2 f(x^0) - \sum_{i=1}^m \pi_i^0 \nabla^2 g_i(x^0) \quad (5.10)$$

The coefficient matrix of (5.9) is called the KKT-matrix. A unique solution of (5.9) exists if the KKT-matrix is non-singular. Otherwise no solution or several solutions may exist. Non-singularity of the KKT-matrix is closely related to the linear independence constraint qualification and the second order sufficiency condition. This relation is stated in the following proposition.

**Proposition 5.2.6**

The KKT-matrix

$$\begin{bmatrix} \nabla_{xx}^2 \mathcal{L}(x^0, \pi^0) & -\nabla g(x^0) \\ -\nabla g(x^0)' & 0 \end{bmatrix} \quad (5.11)$$

is non-singular if and only if  $x^0$  satisfy LICQ and

$$p' \nabla_{xx} \mathcal{L}(x^0, \pi^0) p > 0 \quad \forall p \in V(x^0) \setminus \{0\} \quad (5.12)$$

*Proof.* See Nocedal and Wright (1999) □

Consequently, under the assumption that the linear independence constraint qualification holds, satisfaction of the second order sufficiency condition is detected by monitoring the singularity of the KKT-matrix during its factorization. The second order sufficiency condition is equivalent to positive definiteness of the Hessian matrix projected on the null space of  $\nabla g(x^0)$ .

Through this overview of the optimality conditions, it is clear that if the linear independence constraint qualification and the second order sufficiency condition hold at the points generated by Newton's algorithm, then a local minimizer of (5.1) may be located by solving (5.3) using Newton's method.

### 5.2.2 Newton's Method and SQP

An alternative conceptualization of the Newton procedure for computation of the step direction is to consider the quadratic program

$$\min_{\Delta x \in \mathbb{R}^n} \quad \phi = \frac{1}{2} \Delta x' \nabla_{xx}^2 \mathcal{L}(x^0, \pi^0) \Delta x + \nabla_x \mathcal{L}(x^0, \pi^0)' \Delta x \quad (5.13a)$$

$$s.t. \quad g(x^0) + \nabla g(x^0)' \Delta x = 0 \quad (5.13b)$$

If the second order sufficiency condition holds at  $(x^0, \pi^0)$  then the KKT-system (5.9) is a necessary and sufficient conditions for  $\Delta x$  being a unique global minimizer of (5.13). Furthermore, if the linear independence constraint qualification is satisfied then the KKT-matrix (5.11) is non-singular. Otherwise, the quadratic program is either infeasible due to inconsistent constraints or some constraints are superfluous. The first order conditions solved by Newton's method may be obtained as solution of a sequence of quadratic programs (5.13) and is therefore called sequential quadratic programming.

In practice sequential quadratic programming algorithms do not compute the search direction by solving (5.13). Rather, they solve the quadratic program

$$\min_{\Delta x \in \mathbb{R}^n} \quad \phi = \frac{1}{2} \Delta x' W \Delta x + \nabla f(x^0)' \Delta x \quad (5.14a)$$

$$s.t. \quad g(x^0) + \nabla g(x^0)' \Delta x = 0 \quad (5.14b)$$

in which  $W$  is some approximation of the Hessian of the Lagrangian. A KKT-point of the quadratic program (5.14) is characterized by the linear system of equations

$$\begin{bmatrix} W & -\nabla g(x^0) \\ -\nabla g(x^0)' & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \mu \end{bmatrix} = - \begin{bmatrix} \nabla f(x^0) \\ -g(x^0) \end{bmatrix} \quad (5.15)$$

If  $W = \nabla_{xx}^2 \mathcal{L}(x^0, \pi^0)$  and  $\Delta \pi$  is defined by

$$\mu = \pi^0 + \Delta \pi \quad (5.16)$$

then equation (5.15) is identical with equation (5.9). Consequently, Newton's method for solution of (5.3) can be regarded as a sequential quadratic programming method in which the search direction is generated by (5.14) and the

associated step in the Lagrange multipliers is  $\Delta\pi = \mu - \pi^0$ .  $\mu$  is the Lagrange multipliers associated with the quadratic program (5.14).

$W$  is usually not the exact Hessian of the Lagrangian. It is a secant approximation. For sequential quadratic programming a common approximation is the modified BFGS update due to Powell (1977). In this update  $s$  and  $y$  are computed by

$$s = x - x^0 \quad (5.17a)$$

$$y = \nabla_x \mathcal{L}(x, \pi) - \nabla_x \mathcal{L}(x^0, \pi) \quad (5.17b)$$

in which  $\pi$  is the most recently computed Lagrange multipliers of (5.1). Subsequently,  $\theta$  is computed by

$$\theta = \begin{cases} 1 & s'y \geq 0.2s'Ws \\ \frac{0.8s'Ws}{s'Ws - s'y} & s'y < 0.2s'Ws \end{cases} \quad (5.18)$$

and  $r$  is defined as

$$r = \theta y + (1 - \theta)Ws \quad (5.19)$$

such that the secant approximation of the Hessian of the Lagrangian,  $W$ , may be updated according to

$$W \leftarrow W - \frac{Wss'W}{s'Ws} + \frac{rr'}{s'r} \quad (5.20)$$

For unconstrained optimization, positive definiteness of the Hessians approximated by the BFGS update is guaranteed if the curvature condition,  $s'y > 0$ , is satisfied. This condition is incorporated in the line search algorithm such that it is always satisfied (c.f. Dennis and Schnabel, 1996). However, for constrained optimization situations exist in which this condition is not satisfied for any value of the step length. To overcome this difficulty Powell modified the BFGS update procedure at the price of loosing the secant property. Powell's modification ensures that the curvature condition is satisfied, i.e.  $s'r \geq 0.2s'Ws > 0$  for  $s \neq 0$ . When the first matrix,  $W$ , is positive definite, this update ensures that all subsequent Hessian approximations,  $W$ , are positive definite. Hence, if  $\nabla g(x)$  has full column rank, i.e. satisfy the linear independence constraint qualification, for all  $x$  generated by the SQP algorithm, then the search direction implicitly defined by equation (5.15) exists and is unique. It is important to notice that sequential quadratic programming algorithm with a modified BFGS update requires evaluation of the functions and their first order derivatives but not their second order derivatives.

To ensure convergence, the full step  $(\Delta x, \Delta\pi)$  computed by solution of (5.15) is not necessarily taken.  $(\Delta x, \Delta\pi)$  is used as a search direction and the step taken is given by

$$\begin{bmatrix} x \\ \pi \end{bmatrix} = \begin{bmatrix} x^0 \\ \pi^0 \end{bmatrix} + \alpha \begin{bmatrix} \Delta x \\ \Delta\pi \end{bmatrix} = \begin{bmatrix} x^0 \\ \pi^0 \end{bmatrix} + \alpha \begin{bmatrix} \Delta x \\ \mu - \pi^0 \end{bmatrix} \quad (5.21)$$

in which  $\alpha \in (0, 1]$  is the step length. The procedure for selecting  $\alpha$  is described in the next section when the major steps of the entire algorithm have been presented.

Convergence to a local minimum is checked using the KKT-tolerance due to Chen and Stadtherr (1984). The KKT-tolerance is a weighted sum of possible objective function improvement and constraint violations. This convergence criterion is

$$|\nabla f(x^0)' \Delta x| + \sum_{i=1}^m |\mu_i g_i(x^0)| \leq \epsilon \quad (5.22)$$

and may be checked immediately after computation of the search direction. The rationale for this convergence criterion is explained in the next proposition.

**Proposition 5.2.7**

Assume that LICQ holds at  $x^0$ . Let the approximate Hessian matrix,  $W$ , be positive definite. Let  $(\Delta x, \mu)$  be the solution of (5.15). Then

$$|\nabla f(x^0)' \Delta x| + \sum_{i=1}^m |\mu_i g_i(x^0)| = 0 \quad (5.23)$$

implies  $\Delta x = 0$  and

$$\nabla f(x^0) - \nabla g(x^0)\mu = 0 \quad (5.24a)$$

$$g(x^0) = 0 \quad (5.24b)$$

*Proof.* (5.23) implies  $\nabla f(x^0)' \Delta x = 0$  and  $\mu_i g_i(x^0) = 0$  for  $i = 1, 2, \dots, m$ . Combined with (5.15) these relations give

$$\begin{aligned} 0 &= \nabla f(x^0)' \Delta x = - (W \Delta x - \nabla g(x^0)\mu)' \Delta x \\ &= -\Delta x' W \Delta x + \mu' \nabla g(x^0)' \Delta x \\ &= -\Delta x' W \Delta x - \underbrace{\mu' g(x^0)}_{=0} = -\Delta x' W \Delta x \end{aligned} \quad (5.25)$$

which implies  $\Delta x = 0$  as  $W$  is positive definite. Consequently, using  $\Delta x = 0$ , the first block row of (5.15) yields

$$-\nabla g(x^0)\mu = -\nabla f(x^0) \Leftrightarrow \nabla f(x^0) - \nabla g(x^0)\mu = 0 \quad (5.26)$$

The second block row of (5.15) yields

$$g(x^0) = -\nabla g(x^0)' \Delta x = -\nabla g(x^0)' 0 = 0 \quad (5.27)$$

This establishes the claimed relations.  $\square$

By proposition 5.2.7, it is clear that a KKT-point  $(x^*, \pi^*)$  satisfying (5.3) may be established from the triple  $(x^0, \Delta x, \mu)$  satisfying (5.23) by  $(x^*, \pi^*) = (x^0, \mu)$ . Therefore, (5.22) may be used to detect whether a KKT-point has been found.

This criterion has the same unit as the objective function and it gives an approximate measure for the number of correct digits to be expected in the numerical value of the objective function. Gill *et al.* (1997) implement an alternative convergence test. They test directly whether the KKT-conditions are satisfied at the current point.

The major steps in the sequential quadratic programming method for solution of the nonlinear program (5.1) are stated in the following algorithm.

**Algorithm 6 (Infeasible Path SQP with BFGS)**

Given  $(x^0, \pi^0)$

1. Evaluate the functions and gradients at the initial point: Calculate  $f^0 = f(x^0)$ ,  $g^0 = g(x^0)$ ,  $c = \nabla f(x^0)$ , and  $A = \nabla g(x^0)$ .
2. Choose an initial Hessian approximation:  $W$  is chosen as some symmetric positive definite matrix. This is an initial estimate for the Hessian matrix.
3. Compute  $(\Delta x, \mu)$  by solving the optimality conditions (5.15) of (5.14):

$$\begin{bmatrix} W & -A \\ -A' & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \mu \end{bmatrix} = - \begin{bmatrix} c \\ -g^0 \end{bmatrix} \quad (5.28)$$

4. Compute the termination criteria and check for convergence. If

$$|c' \Delta x| + \sum_{i=1}^m |\mu_i g_i^0| \leq \epsilon \quad (5.29)$$

terminate with  $(x^0, \mu)$  as the optimal solution. The optimal objective function value is  $f^0$ .

5. Compute  $\alpha$  by a line search using some merit function. The result is  $\alpha$ ,  $x = x^0 + \alpha \Delta x$ ,  $f = f(x)$ , and  $g = g(x)$ . Compute  $\pi = \pi^0 + \alpha(\mu - \pi^0)$ .
6. Compute

$$\nabla_x \mathcal{L}(x^0, \pi) = c - A\pi \quad (5.30)$$

7. Evaluate the gradients at the new point: Calculate  $c \leftarrow \nabla f(x)$  and  $A \leftarrow \nabla g(x)$ .
8. Compute

$$\nabla_x \mathcal{L}(x, \pi) = c - A\pi \quad (5.31)$$

9. Update the Hessian matrix by the modified BFGS recursion. Compute

$$s = x - x^0 \quad (5.32a)$$

$$y = \nabla_x \mathcal{L}(x, \pi) - \nabla_x \mathcal{L}(x^0, \pi) \quad (5.32b)$$

and

$$\theta = \begin{cases} 1 & s'y \geq 0.2s'Ws \\ \frac{0.8s'Ws}{s'Ws - s'y} & s'y < 0.2s'Ws \end{cases} \quad (5.33)$$

as well as

$$r = \theta y + (1 - \theta)Ws \quad (5.34)$$

The updated Hessian is computed as

$$W \leftarrow W - \frac{Wss'W}{s'Ws} + \frac{rr'}{s'r} \quad (5.35)$$

10. Update the old values:  $x^0 \leftarrow x$ ,  $\pi^0 \leftarrow \pi$ ,  $f^0 \leftarrow f$ ,  $g^0 \leftarrow g$ . Goto (3).

A significant part of the computational cost of the algorithm is concerned with solution of the KKT-system (5.28). For some problems evaluation of the objective function, evaluation of the constraint functions as well as evaluation of their gradients constitute a significant computation cost that cannot be neglected. This is the case for the optimal control problem. Some merit functions compute  $\pi$  in step (6). However, as described in the next subsection this is not done in the line search algorithm using Powell's  $l_1$ -penalty function.

### 5.2.3 Merit Function and Step Length

The solution generated by the infeasible path SQP algorithm is only guaranteed to be feasible within a certain tolerance at the termination of the algorithm. At intermediate steps the solution is typically infeasible. Therefore the objective function cannot be used to decide the step length,  $\alpha$ . Instead a merit function which is some combination of the objective function and the constraint violation is used. The actual merit function used in selecting the step length is important for the convergence rate of the SQP algorithm. Powell (1977) suggested an  $l_1$  merit function, while Schittkowski (1981a,b), Fletcher (1987), and Gill *et al.* (1997) have proposed augmented Lagrangian merit functions.

For simplicity, the  $l_1$ -merit function due to Powell (1977) is considered

$$P(x, \sigma) = f(x) + \sum_{i=1}^m \sigma_i |g_i(x)| \quad (5.36)$$

in which the penalty parameters,  $\sigma_i$ , are chosen as

$$\sigma_i \leftarrow \max \left\{ |\mu_i|, \frac{1}{2}(\sigma_i + |\mu_i|) \right\} \quad i = 1, 2, \dots, m \quad (5.37)$$

except at the first iteration at which they are set equal to the numerical value of the Lagrange multipliers, i.e.  $\sigma_i = |\mu_i|$ . The merit function of the point

$$\begin{pmatrix} x \\ \pi \end{pmatrix} = \begin{pmatrix} x^0 \\ \pi^0 \end{pmatrix} + \alpha \begin{pmatrix} \Delta x \\ \mu - \pi^0 \end{pmatrix} \quad (5.38)$$

may be stated as a function of  $\alpha$

$$T(\alpha) = P(x^0 + \alpha \Delta x, \sigma) \quad (5.39)$$

In an exact line search procedure  $\alpha$  is chosen such that  $T(\alpha)$  is minimized. However, the exact line search is seldom used as the computational cost in terms of function evaluations is unjustified when it is compared to an inexact (or soft) line search in which  $\alpha$  is selected such that a sufficient decrease in  $T(\alpha)$  is obtained. The sufficient decrease in the function  $T(\alpha)$  is measured by the *Armijo* condition

$$T(\alpha) \leq T(0) + c_1 \alpha \frac{dT}{d\alpha}(0) \quad (5.40)$$

in which  $c_1 \in (0, 1)$ . Nocedal and Wright (1999) recommend  $c_1 = 10^{-4}$  while Powell (1977) used  $c_1 = 0.1$ . The Armijo condition itself is not sufficient to guarantee that the algorithm makes sufficient progress as it does not rule out small value of the step length,  $\alpha$ . To rule out too small step lengths,  $\alpha$  may be selected by backtracking, by using a Wolfe condition, or by using a Goldstein condition (c.f. Nocedal and Wright, 1999).

The computation of the step length,  $\alpha$ , using the Armijo condition (5.40) and backtracking in an inexact line search procedure is facilitated by the following lemma

**Lemma 5.2.8**

Let  $\Delta x$  be a solution of (5.14) and let  $T : [0, 1] \mapsto \mathbb{R}$  be defined by (5.39) and (5.36). Then

$$T(\alpha) = f(x^0 + \alpha\Delta x) + \sum_{i=1}^m \sigma_i |g_i(x^0 + \alpha\Delta x)| \quad (5.41)$$

and

$$T(0) = P(x^0, \sigma) = f(x^0) + \sum_{i=1}^m \sigma_i |g_i(x^0)| \quad (5.42a)$$

$$T'(0) = \frac{dT}{d\alpha}(0) = \nabla f(x^0)' \Delta x - \sum_{i=1}^m \sigma_i |g_i(x^0)| \quad (5.42b)$$

*Proof.* The expressions for  $T(\alpha)$  and  $T(0)$  follows straightforwardly. To deduce the expression for the derivative,  $T'(0)$ , notice that the gradient

$$\nabla |g_i(x)| = \begin{cases} \nabla g_i(x) & g_i(x) \geq 0 \\ -\nabla g_i(x) & g_i(x) \leq 0 \end{cases} \quad (5.43)$$

is not uniquely defined for  $g_i(x) = 0$ . This implies that the line search function  $T(\alpha)$  is not in general differentiable everywhere. A special case is  $\alpha = 0$ .  $\Delta x$  is constructed such that it satisfies (5.14b) which is equivalent to

$$\nabla g_i(x^0)' \Delta x = -g_i(x^0) \quad (5.44)$$

Hence

$$\begin{aligned} \nabla |g_i(x^0)|' \Delta x &= \begin{cases} \nabla g_i(x^0)' \Delta x & g_i(x^0) \geq 0 \\ -\nabla g_i(x^0)' \Delta x & g_i(x^0) \leq 0 \end{cases} \\ &= \begin{cases} -g_i(x^0) & g_i(x^0) \geq 0 \\ g_i(x^0) & g_i(x^0) \leq 0 \end{cases} \\ &= -|g_i(x^0)| \end{aligned} \quad (5.45)$$



For  $\alpha = 0$  the derivative of the line search function becomes

$$\begin{aligned}
 T'(0) &= \frac{dT}{d\alpha}(0) = \Delta x' \nabla_x P(x^0, \sigma) \\
 &= \Delta x' \left( \nabla f(x^0) + \sum_{i=1}^m \sigma_i \nabla |g_i(x^0)| \right) \\
 &= \nabla f(x^0)' \Delta x - \sum_{i=1}^m \sigma_i \nabla |g_i(x^0)|' \Delta x \\
 &= \nabla f(x^0)' \Delta x - \sum_{i=1}^m \sigma_i |g_i(x^0)|
 \end{aligned} \tag{5.46}$$

□

The inexact line search procedure used, initially attempts a full step ( $\alpha_1 = 1$ ) and terminates if the sufficient decrease condition (5.40) is satisfied. Otherwise, it constructs a quadratic interpolation based on  $T(0)$ ,  $T'(0)$ , and  $T(1)$ . The step length,  $\alpha_2$ , is selected as the minimizer of this quadratic interpolation in the interval  $\alpha_2 \in [0.1\alpha_1, 1]$ . The line search procedure is terminated if the sufficient decrease condition is satisfied at this new value of the step length. Otherwise, a cubic interpolation based on the points  $T(0)$ ,  $T'(0)$ ,  $T(\alpha_1)$ , and  $T(\alpha_2)$  is constructed. The step length,  $\alpha$  in the interval  $[0.1\alpha_2, 0.5\alpha_2]$  that minimizes the cubic interpolation of  $T(\alpha)$  is selected as the new step length. This process with cubic interpolation is continued based on the two most recent step lengths,  $\alpha_1$  and  $\alpha_2$ , until the sufficient decrease condition (5.40) is satisfied. The details of this procedure are described by Nocedal and Wright (1999) as well as by Dennis and Schnabel (1996).

The quadratic interpolation of  $T(\alpha)$  is

$$\phi(\alpha) = (T(1) - T(0) - T'(0))\alpha^2 + T'(0)\alpha + T(0) \tag{5.47}$$

and its unconstrained minimizer is

$$\alpha_{\min} = \frac{T'(0)}{2(T(0) + T'(0) - T(1))} \tag{5.48}$$

The step length  $\alpha_2$  is then computed by

$$\alpha_2 = \max\{0.1\alpha_1, \alpha_{\min}\} \tag{5.49}$$

in which  $\alpha_1 = 1$ . The cubic interpolation of  $T(\alpha)$  is

$$\phi(\alpha) = a\alpha^3 + b\alpha^2 + T'(0)\alpha + T(0) \tag{5.50}$$

in which

$$\begin{bmatrix} a \\ b \end{bmatrix} = \frac{1}{\alpha_1 - \alpha_2} \begin{bmatrix} \frac{1}{\alpha_1^2} & -\frac{1}{\alpha_2^2} \\ -\frac{\alpha_2}{\alpha_1} & \frac{\alpha_1}{\alpha_2} \end{bmatrix} \begin{bmatrix} T(\alpha_1) - T'(0)\alpha_1 - T(0) \\ T(\alpha_2) - T'(0)\alpha_2 - T(0) \end{bmatrix} \tag{5.51}$$

The unconstrained minimum of (5.50) is

$$\alpha_{\min} = \frac{-b + \sqrt{b^2 - 3aT'(0)}}{3a} \quad (5.52)$$

and the new attempted step length is

$$\alpha = \begin{cases} 0.1\alpha_2 & \alpha_{\min} \leq 0.1\alpha_2 \\ 0.5\alpha_2 & \alpha_{\min} \geq 0.5\alpha_2 \\ \alpha_{\min} & \text{otherwise} \end{cases} \quad (5.53)$$

In the construction of the cubic interpolation,  $\alpha_2$  is the most recent step length attempted and  $\alpha_1$  is the second most recent step length attempted. Based on this interpretation of  $\alpha_1$  and  $\alpha_2$  the computation of  $\alpha$  based on cubic interpolation may be continued until a value of  $\alpha$  satisfies the sufficient decrease condition (5.40).

The major steps in an inexact backtracking line search procedure using Powell's  $l_1$ -penalty function are summarized in the following algorithm.

**Algorithm 7 (Line Search with Powell's  $l_1$ -Penalty)**

Let  $f(x^0)$ ,  $g(x^0)$ ,  $\nabla f(x^0)$ ,  $x^0$ ,  $\Delta x$ , and  $\sigma$  be given. Let  $c_1 = 10^{-4}$ .

1. Compute  $\eta_0 = \sum_{i=1}^m \sigma_i |g_i(x^0)|$ ,  $T(0) = f(x^0) + \eta_0$ , and  $T'(0) = \nabla f(x^0)' \Delta x_0 - \eta_0$ . Set  $\alpha_1 = 1$ .
2. Compute  $x = x^0 + \Delta x$ . Evaluate  $f = f(x)$  and  $g = g(x)$ . Compute  $T(\alpha_1) = f + \sum_{i=1}^m \sigma_i |g_i|$ .
3. If  $T(\alpha_1) \leq T(0) + c_1 T'(0)$  then stop with  $\alpha = \alpha_1$ .
4. Compute  $\alpha_{\min}$  by (5.48) and  $\alpha_2$  by (5.49).
5.  $x = x^0 + \alpha_2 \Delta x$ . Evaluate  $f = f(x)$  and  $g = g(x)$ . Compute  $T(\alpha_2) = f + \sum_{i=1}^m \sigma_i |g_i|$ .
6. If  $T(\alpha_2) \leq T(0) + c_1 \alpha_2 T'(0)$  then stop with  $\alpha = \alpha_2$ .
7. Compute  $a$  and  $b$  by (5.51). If  $a = 0$  compute  $\alpha_{\min} = \frac{-T'(0)}{b}$ . Otherwise compute  $\alpha_{\min}$  by (5.52). Compute  $\alpha$  by (5.53).
8.  $x = x^0 + \alpha \Delta x$ . Evaluate  $f = f(x)$  and  $g = g(x)$ . Compute  $T(\alpha) = f + \sum_{i=1}^m \sigma_i |g_i|$ .
9. If  $T(\alpha) \leq T(0) + c_1 \alpha T'(0)$  then stop.
10. Set  $\alpha_1 = \alpha_2$ ,  $\alpha_2 = \alpha$ ,  $T(\alpha_1) = T(\alpha_2)$ , and  $T(\alpha_2) = T(\alpha)$ . Go to step (7).

The algorithm returns  $\alpha$ ,  $x = x^0 + \alpha \Delta x$ ,  $f = f(x)$  and  $g = g(x)$ .

Algorithm 6 and 7 in conjunction can be used to compute a local minimizer of (5.1). These algorithms constitute a line search SQP algorithm for equality constrained nonlinear programming.

### 5.3 SQP for Nonlinear Optimal Control

The unconstrained nonlinear optimal control problem considered is a continuous time Bolza problem

$$\min_{x(t), u(t)} \psi = \int_{t_0}^{t_f} l(x(t), u(t), t) dt + L(x(t_f), t_f) \quad (5.54a)$$

$$s.t. \quad \dot{x}(t) = h(x(t), u(t), t) \quad (5.54b)$$

$$x(t_0) = x_0 \quad (5.54c)$$

The objective is to compute the control function  $u(t) = u(t, x_0)$  such that some performance integral is minimized. This integral may represent production costs or be a least squares objective aimed at tracking some specified trajectory. The evolution of  $x(t)$  given  $u(t)$  and  $x_0$  is described by a nonlinear system of ordinary differential equations (5.54b).

#### 5.3.1 Solution Methods

The history of the optimal control problem (5.54) and its constrained relatives is described by Polak (1973) and more recently by Bryson (1996). Polak (1973) classifies the major methods for solution of the optimal control problem as gradient methods, Newton methods, Gauss-Newton methods, and conjugate gradient methods. He discusses extensively the relations between the unconstrained optimization problem and the unconstrained optimal control problem. Pesch (1994) provides a practical guide to the solution of optimal control problems, while Kraft (1985, 1994) describes the discretization of (5.54) and practical solution by nonlinear programming. The main categories of solution methods for the optimal control problem (5.54) are summarized in table 5.1. This classification is different from the classification provided by Polak (1973). Polak (1973) classifies according to the numerical method applied when the optimal control problem is converted to an unconstrained optimization problem. In table 5.1, the methods are classified according to the optimality conditions used for solving the unconstrained optimal control problem. The solution methods are classified as direct and indirect methods. The direct methods transform the infinite dimensional optimization problem into a finite dimensional nonlinear program, while the indirect methods formulate optimality conditions of the continuous-time optimal control problem (5.54) and compute the solution based on these conditions (c.f. Binder *et al.*, 2001a). The indirect methods comprise algorithms based on Pontryagin's maximum principle (PMP) and algorithms based on solution of the Hamilton-Jacobi-Bellman (HJB) partial differential equation.

Using Pontryagin's maximum principle (PMP), the indirect methods derive the optimality conditions for the continuous-time problem and compute the solution using these conditions. The conditions developed are the Euler-Lagrange

**Table 5.1.** Methodologies for solution of continuous-time optimal control problems (cont. = continuous, param. = parameterized, PMP = Pontryagin's maximum principle, HJB = Hamilton-Jacobian-Bellman equation, BCI = boundary condition iteration, CVP = control vector parameterization, CVI = control vector iteration, NLP = nonlinear program, DP = dynamic programming).

| Numerical solution                 | Direct                | Indirect PMP             | Indirect HJB          |
|------------------------------------|-----------------------|--------------------------|-----------------------|
| States - cont.<br>Inputs - cont.   | -                     | Shooting method (BCI)    | -                     |
| States - cont.<br>Inputs - param.  | Feasible path (CVP)   | Gradient method (CVI)    | -                     |
| States - param.<br>Inputs - param. | Infeasible path (NLP) | State and adjoint param. | Dynamic program. (DP) |

differential equations and the controls are obtained by application of the maximum principle (c.f. Hartl *et al.*, 1995; Bryson and Ho, 1975). Solution procedures using the maximum principle of Pontryagin introduce the Hamiltonian of (5.54) defined by

$$\mathcal{H}(x(t), u(t), \pi(t), t) = l(x(t), u(t), t) - \pi(t)'h(x(t), u(t), t) \quad (5.55)$$

Necessary conditions for optimality of a trajectory,  $(x^*(t), u^*(t))$ , can be expressed by the differential equations

$$\dot{x}^*(t) = -\nabla_{\pi}\mathcal{H}(x^*(t), u^*(t), \pi^*(t), t) \quad (5.56a)$$

$$\dot{\pi}^*(t) = \nabla_x\mathcal{H}(x^*(t), u^*(t), \pi^*(t), t) \quad (5.56b)$$

with boundary conditions

$$x^*(t_0) = x_0 \quad (5.57a)$$

$$\pi^*(t_f) = \nabla_x L(x^*(t_f), t_f) \quad (5.57b)$$

and

$$u^*(t) = \arg \min_{u(t)} \mathcal{H}(x^*(t), u(t), \pi^*(t), t) \quad (5.58)$$

The latter condition is often enforced by the necessary optimality condition

$$\nabla_u \mathcal{H}(x^*(t), u^*(t), \pi^*(t), t) = 0 \quad (5.59)$$

The boundary condition iteration (BCI) procedure for solution of the optimal control problem (5.54) computes the optimal solution by computing  $\pi^*(t_0) = \pi_0$

such that the boundary condition (5.57b) is satisfied.  $\pi^*(t_0) = \pi_0$  is found iteratively by a shooting procedure integrating (5.56) forward and solving (5.58) along the obtained trajectory.  $\pi^*(t_0) = \pi_0$  is adjusted by a Newton or gradient method until the boundary condition (5.57b) is satisfied. Given the optimal initial adjoint variable,  $\pi^*(t_0)$ , satisfying the boundary condition (5.57b) the optimal trajectory may be computed by integrating (5.56) forward and solving (5.58) along this trajectory. This procedure may fail for unstable systems due to divergence or the large sensitivity to the initial conditions.

By the gradient method, which is also called the control vector iteration (CVI) method, the control vector,  $u(t)$ , is parameterized. Using this parameterization the Hamiltonians are minimized subject to the dynamic constraints and the boundary conditions. Given an initial guess of the decision variables  $u(t)$  in terms of their parameterization, the equation

$$\begin{aligned}\dot{x}^*(t) &= -\nabla_{\pi} \mathcal{H}(x^*(t), u^*(t), \pi^*(t), t) \\ &= h(x^*(t), u^*(t), t)\end{aligned}\tag{5.60}$$

is integrated forward. Subsequently, the adjoint equation (5.56b) is integrated backward. Having a trajectory  $(x(t), \pi(t))$ , some gradient based optimization algorithm is used to adjust the decision variables,  $u(t)$ , such that they ultimately converge to the optimal solution,  $u^*(t)$ , and satisfy (5.58). However, this procedure may as the boundary vector iteration method fail for unstable systems due to divergence and high sensitivity of the decision variables.

By the state and adjoint parameterization, the differential equations (5.56) are discretized and all equations constituting the necessary conditions are solved simultaneously using a Newton procedure. At each iteration a very large system of equations must be solved. However, due to this discretization in which controls, states and adjoint variables are computed simultaneously, this method may be applicable to stable as well as unstable systems.

Another classical indirect method for solution of the optimal control problem is by solution of the Hamilton-Jacobi-Bellman (HJB) equation. By this procedure optimality is characterized by the value function,  $V(x^*(t), t)$  of (5.54) and by application of Bellman's *principle of optimality*. As such the Hamilton-Jacobi-Bellman procedure may be regarded as a continuous-time version of dynamic programming. The optimal trajectory,  $(x^*(t), u^*(t))$  can be obtained by solution of the HJB-equation

$$\begin{aligned}\frac{\partial V(x^*(t), t)}{\partial t} &= -\min_u \{l(x^*(t), u(t), t) \\ &\quad + \nabla_x V(x^*(t), t)' h(x^*(t), u(t), t)\}\end{aligned}\tag{5.61}$$

along with its boundary condition

$$V(x^*(t_f), t_f) = L(x^*(t_f), t_f)\tag{5.62}$$

By this procedure the optimal  $u^*(t)$  is computed as a function of  $x^*(t)$ . Specific values for  $u^*(t)$  is obtained using the initial condition  $x^*(t_0) = x_0$ . However,

due to the curse of dimensionality this is not a practical approach and few realistic applications have been reported. An exception is the case of linear models in which an analytic solution is available.

The direct methods are the most practical therefore also the commonly applied methods for approximate solution of the nonlinear optimal control problem (5.54). By the direct methods the infinite dimensional optimization problem (5.54) is converted to a finite dimensional nonlinear program by parameterization of the controls,  $u(t)$ . The control vector parameterization (CVP) algorithm parameterizes the controls and solves the differential equations (5.54b) exactly at each iteration of the optimization algorithm. The control vector parameterization method is also referred to as a feasible path method as the differential equations are satisfied at all iterations. Furthermore, as the integration and optimization are conducted sequentially, the control vector parameterization method is also called a sequential method. In contrast, by the infeasible path method the differential equations (5.54b) are discretized and the resulting nonlinear program (NLP) is solved. Therefore, by the infeasible path method the integration and optimization is conducted simultaneously. Satisfaction of the differential equations are only guaranteed at the optimal solution but not at intermediate iterates. As by the indirect sequential approaches, the control vector parameterization method may suffer from high sensitivity and divergence when applied to unstable systems. Such problems do not occur using the infeasible path method as optimization and integration are conducted simultaneously.

### 5.3.2 Transcription in the Direct Methods

The basic idea in the direct methods is to transcribe the infinite dimensional optimization problem (5.54) into a finite dimensional optimization problem. This is achieved by approximating the optimal control  $u(t)$  by some finite dimensional function. In this paper, the optimal control is approximated by a piecewise constant function. This is also known as the zero-order-hold approximation. The integral in (5.54a) and the system of nonlinear differential equations (5.54b) are approximated by difference equations using a numerical integration method. For simplicity and to illustrate the principles, the explicit Euler integration method is used in this paper. However, this is not a recommendation of the explicit Euler method as the most efficient method. By these approximation the finite dimensional approximation of (5.54) becomes

$$\min_{x,u} f(x,u) = \sum_{k=0}^{N-1} f_k(x_k, u_k) + f_N(x_N) \quad (5.63a)$$

$$s.t. \quad x_{k+1} = H_k(x_k, u_k), \quad k = 0, 1, \dots, N-1 \quad (5.63b)$$

in which  $x_0$  is a parameter and the decision variables are  $x = \{x_k\}_{k=1}^N$  and  $u = \{u_k\}_{k=0}^{N-1}$ . To emphasize the fact that  $x_0$  is a parameter and not a decision

variable, (5.63) may be expressed as

$$\min_{x,u} \quad f = f_0(u_0) + \sum_{k=1}^{N-1} f_k(x_k, u_k) + f_N(x_N) \quad (5.64a)$$

$$s.t. \quad x_1 = H_0(u_0) \quad (5.64b)$$

$$x_{k+1} = H_k(x_k, u_k), \quad k = 1, 2, \dots, N-1 \quad (5.64c)$$

in which  $f_0(u_0) = f_0(x_0, u_0)$  and  $H_0(u_0) = H_0(x_0, u_0)$ . The equivalent mathematical programs (5.63) and (5.64) representing the discrete time Bolza problem could be solved using directly the sequential quadratic programming algorithm described in section 5.2. However, the nonlinear discrete time control (5.64) problem has a special structure that may be utilized in its efficient solution.

The purpose of this section is to describe the computations used in construction of the discrete time approximation (5.64) and the efficient solution of (5.64) by using a modification of the sequential quadratic programming algorithm described in section 5.2.

### 5.3.3 The Lagrangian and Partial Separability

The Lagrangian,  $\mathcal{L} = \mathcal{L}(x, u, \pi)$ , of (5.64) is

$$\begin{aligned} \mathcal{L} &= f_0(u_0) + \sum_{k=1}^{N-1} f_k(x_k, u_k) + f_N(x_N) \\ &\quad - \pi'_0(H_0(u_0) - x_1) \\ &\quad - \sum_{k=1}^{N-1} \pi'_k(H_k(x_k, u_k) - x_{k+1}) \\ &= \mathcal{L}_0(u_0, \pi) + \sum_{k=1}^{N-1} \mathcal{L}_k(x_k, u_k, \pi) + \mathcal{L}_N(x_N, \pi) \end{aligned} \quad (5.65)$$

in which  $\mathcal{L}_0 = \mathcal{L}_0(u_0, \pi)$ ,  $\mathcal{L}_k = \mathcal{L}_k(x_k, u_k, \pi)$ , and  $\mathcal{L}_N = \mathcal{L}_N(x_N, \pi)$  are defined by

$$\mathcal{L}_0 = f_0(u_0) - \pi'_0 H_0(u_0) \quad (5.66a)$$

$$\mathcal{L}_k = f_k(x_k, u_k) - \pi'_k H_k(x_k, u_k) + \pi'_{k-1} x_k \quad (5.66b)$$

$$\mathcal{L}_N = f_N(x_N) + \pi'_{N-1} x_N \quad (5.66c)$$

The first order necessary optimality conditions specialized to (5.64) are

$$\nabla_{u_0} \mathcal{L}(x, u, \pi) = \nabla_{u_0} \mathcal{L}_0(u_0, \pi) = 0 \quad (5.67a)$$

$$\nabla_{x_k} \mathcal{L}(x, u, \pi) = \nabla_{x_k} \mathcal{L}_k(x_k, u_k, \pi) = 0 \quad (5.67b)$$

$$\nabla_{u_k} \mathcal{L}(x, u, \pi) = \nabla_{u_k} \mathcal{L}_k(x_k, u_k, \pi) = 0 \quad (5.67c)$$

$$\nabla_{x_N} \mathcal{L}(x, u, \pi) = \nabla_{x_N} \mathcal{L}_N(x_N, \pi) = 0 \quad (5.67d)$$

as well as

$$\nabla_{\pi_0} \mathcal{L}(x, u, \pi) = -H_0(u_0) + x_1 = 0 \quad (5.68a)$$

$$\nabla_{\pi_k} \mathcal{L}(x, u, \pi) = -H_k(x_k, u_k) + x_{k+1} = 0 \quad (5.68b)$$

The conditions (5.67) may also be expressed as

$$\nabla_{u_0} \mathcal{L}_0 = \nabla_{u_0} f_0(u_0) - \nabla_{u_0} H_0(u_0) \pi_0 = 0 \quad (5.69a)$$

$$\begin{aligned} \nabla_{x_k} \mathcal{L}_k &= \nabla_{x_k} f_k(x_k, u_k) \\ &\quad - \nabla_{x_k} H_k(x_k, u_k) \pi_k + \pi_{k-1} = 0 \end{aligned} \quad (5.69b)$$

$$\begin{aligned} \nabla_{u_k} \mathcal{L}_k &= \nabla_{u_k} f_k(x_k, u_k) \\ &\quad - \nabla_{u_k} H_k(x_k, u_k) \pi_k = 0 \end{aligned} \quad (5.69c)$$

$$\nabla_{x_N} \mathcal{L}_N = \nabla_{x_N} f_N(x_N) + \pi_{N-1} = 0 \quad (5.69d)$$

Let  $z = \{u_0, x_1, u_1, \dots, x_{N-1}, u_{N-1}, x_N\}$ . Then the partial separability of the Lagrangian, i.e.  $\mathcal{L} = \sum_{k=0}^N \mathcal{L}_k$  in which  $\mathcal{L}_0 = \mathcal{L}_0(u_0, \pi)$ ,  $\mathcal{L}_k = \mathcal{L}_k(x_k, u_k, \pi)$ , and  $\mathcal{L}_N = \mathcal{L}_N(x_N, \pi)$ , implies that the Hessian of the Lagrangian,  $\nabla_{zz}^2 \mathcal{L}$ , has a block diagonal structure

$$\nabla_{zz}^2 \mathcal{L} = \begin{bmatrix} R_0 & & & & & \\ & Q_1 & M_1 & & & \\ & M'_1 & R_1 & & & \\ & & & \ddots & & \\ & & & & Q_{N-1} & M_{N-1} \\ & & & & M'_{N-1} & R_{N-1} \\ & & & & & & P_N \end{bmatrix} \quad (5.70)$$

The blocks are symmetric matrices given by the expressions

$$R_0 = \nabla_{u_0, u_0}^2 \mathcal{L}_0 \quad (5.71a)$$

$$\begin{bmatrix} Q_k & M_k \\ M'_k & R_k \end{bmatrix} = \begin{bmatrix} \nabla_{x_k, x_k}^2 \mathcal{L}_k & \nabla_{x_k, u_k}^2 \mathcal{L}_k \\ \nabla_{u_k, x_k}^2 \mathcal{L}_k & \nabla_{u_k, u_k}^2 \mathcal{L}_k \end{bmatrix} \quad (5.71b)$$

$$P_N = \nabla_{x_N, x_N}^2 \mathcal{L}_N \quad (5.71c)$$

### 5.3.4 Partitioned BFGS Hessian Approximation

The SQP algorithm solves (5.1) using a secant approximation of the Hessian matrix of the Lagrangian. Direct construction of an approximate Hessian for



the nonlinear optimal control problem (5.64) would produce a dense matrix,  $W$ , and not retain the block diagonal structure of the exact Lagrangian (5.70). Therefore, the secant approximation,  $W$ , of the Hessian of the Lagrangian is enforced to have the block diagonal structure by updating each block individually (c.f. Bock and Plitt, 1983; Leineweber, 1995; Bock *et al.*, 2000). Thus the approximate Hessian is enforced to have the structure

$$W = \begin{bmatrix} W_0 & & & & \\ & W_1 & & & \\ & & \ddots & & \\ & & & W_{N-1} & \\ & & & & W_N \end{bmatrix} \quad (5.72)$$

and each diagonal block,  $W_k$ , is computed using Powell's modified BFGS update. The structure is obtained by modification of (5.17). At stage  $k = 0$ ,  $s_0$  and  $y_0$  are computed by

$$s_0 = u_0 - u_0^0 \quad (5.73a)$$

$$y_0 = \nabla_{u_0} \mathcal{L}_0(u_0, \pi) - \nabla_{u_0} \mathcal{L}_0(u_0^0, \pi) \quad (5.73b)$$

while at stage  $k = 1, 2, \dots, N-1$ ,  $s_k$  and  $y_k$  are computed by

$$s_k = \begin{bmatrix} x_k - x_k^0 \\ u_k - u_k^0 \end{bmatrix} \quad (5.74a)$$

$$y_k = \begin{bmatrix} \nabla_{x_k} \mathcal{L}_k(x_k, u_k, \pi) - \nabla_{x_k} \mathcal{L}_k(x_k^0, u_k^0, \pi) \\ \nabla_{u_k} \mathcal{L}_k(x_k, u_k, \pi) - \nabla_{u_k} \mathcal{L}_k(x_k^0, u_k^0, \pi) \end{bmatrix} \quad (5.74b)$$

At the final stage  $k = N$ ,  $s_N$  and  $y_N$  are obtained by

$$s_N = x_N - x_N^0 \quad (5.75a)$$

$$y_N = \nabla_{x_N} \mathcal{L}_N(x_N, \pi) - \nabla_{x_N} \mathcal{L}_N(x_N^0, \pi) \quad (5.75b)$$

For each  $k = 0, 1, \dots, N$ ,  $\theta_k$  defined by

$$\theta_k = \begin{cases} 1 & s'_k y_k \geq 0.2 s'_k W_k s_k \\ \frac{0.8 s'_k W_k s_k}{s'_k W_k s_k - s'_k y_k} & s'_k y_k < 0.2 s'_k W_k s_k \end{cases} \quad (5.76)$$

is used to compute  $r_k$

$$r_k = \theta_k y_k + (1 - \theta_k) W_k s_k \quad (5.77)$$

and each block matrix,  $W_k$ , is obtained by the expression

$$W_k \leftarrow \begin{cases} W_k & \|s_k\|_2 \leq \varepsilon_s \\ W_k - \frac{W_k s_k s'_k W_k}{s'_k W_k s_k} + \frac{r_k r'_k}{s'_k r_k} & \|s_k\|_2 > \varepsilon_s \end{cases} \quad (5.78)$$

in which  $\varepsilon_s = 10^{-4}$  is an empirical constant. The distinction between  $\|s_k\|_2$  greater and less than  $\varepsilon_s$  is necessary as  $s_k$  can be zero without the entire vector  $s = \{s_k\}_{k=0}^N$  being zero. That is, some components, say  $s_k$ , can have converged before the entire vector  $s$  has converged. In accordance with the notation used in (5.70) and (5.71) the Hessian block matrices,  $W_k$ , may be denoted

$$R_0 = W_0 \quad (5.79a)$$

$$\begin{bmatrix} Q_k & M_k \\ M'_k & R_k \end{bmatrix} = W_k \quad k = 1, 2, \dots, N-1 \quad (5.79b)$$

$$P_N = W_N \quad (5.79c)$$

### 5.3.5 The Quadratic Program

The data in the quadratic program (5.14) constructed for computation of the search direction used in the sequential quadratic programming algorithm is based on the approximate Hessian of the Lagrangian, the gradient of the objective function of the nonlinear program, and a first order Taylor expansion of the constraints around some nominal point  $x^0$ .

The same procedure is used to construct a quadratic program for computing the search direction in a SQP algorithm for (5.64). The block diagonal structure of the Hessian matrix and the notation defined by (5.79) as well as the separability of the objective function (5.64a) may be used to obtain the following objective function for the quadratic program used to compute the search direction from  $(x^0, u^0)$ . The objective function in the quadratic program is

$$\begin{aligned} \phi &= \frac{1}{2} \Delta u'_0 R_0 \Delta u_0 + \sum_{k=1}^{N-1} \frac{1}{2} \begin{bmatrix} \Delta x_k \\ \Delta u_k \end{bmatrix}' \begin{bmatrix} Q_k & M_k \\ M'_k & R_k \end{bmatrix} \begin{bmatrix} \Delta x_k \\ \Delta u_k \end{bmatrix} \\ &\quad + \frac{1}{2} \Delta x'_N P_N \Delta x_N \\ &\quad + r'_0 \Delta u_0 + \left( \sum_{k=1}^{N-1} q'_k \Delta x_k + r'_k \Delta u_k \right) + p'_N \Delta x_N \\ &= \frac{1}{2} \Delta u'_0 R_0 \Delta u_0 + r'_0 \Delta u_0 \\ &\quad + \sum_{k=1}^{N-1} \left( \frac{1}{2} \Delta x'_k Q_k \Delta x_k + \Delta x'_k M_k \Delta u_k \right. \\ &\quad \quad \left. + \frac{1}{2} \Delta u'_k R_k \Delta u_k + q'_k \Delta x_k + r'_k \Delta u_k \right) \\ &\quad + \frac{1}{2} \Delta x'_N P_N \Delta x_N + p'_N \Delta x_N \end{aligned} \quad (5.80)$$

in which

$$r_0 = \nabla_{u_0} f(x^0, u^0) = \nabla_{u_0} f_0(u_0^0) \quad (5.81a)$$

$$q_k = \nabla_{x_k} f(x^0, u^0) = \nabla_{x_k} f_k(x_k^0, u_k^0), \quad k \in \mathcal{N} \quad (5.81b)$$

$$r_k = \nabla_{u_k} f(x^0, u^0) = \nabla_{u_k} f_k(x_k^0, u_k^0), \quad k \in \mathcal{N} \quad (5.81c)$$

$$p_N = \nabla_{x_N} f(x^0, u^0) = \nabla_{x_N} f_N(x_N^0) \quad (5.81d)$$

and  $\mathcal{N} = \{1, 2, \dots, N-1\}$ . The quadratic program constructed on basis of (5.64) becomes

$$\min_{\Delta x, \Delta u} \phi = \phi(\Delta x, \Delta u) \quad (5.82a)$$

$$s.t. \quad \Delta x_1 = B'_0 \Delta u_0 + b_0 \quad (5.82b)$$

$$\Delta x_{k+1} = A'_k \Delta x_k + B'_k \Delta u_k + b_k, \quad k \in \mathcal{N} \quad (5.82c)$$

in which the objective function is given by (5.80). Even though the quadratic program (5.82) to a large extent resembles the quadratic program constituting the discrete linear quadratic regulator (DLQR) problem (c.f. Franklin *et al.*, 1998), it has some small but important differences. In the DLQR problem, the constraints are linear while the constraints (5.82b)-(5.82c) are affine due to the terms involving the constants,  $b_k$ . Similarly, the objective function in the DLQR problem contains only quadratic terms, while the objective function (5.80) of (5.82) contains quadratic as well as linear terms. For the continuous-time problem, Mitter (1966) was the first to recognize that to solve the optimal control problem by a Newton-Raphson method, the linear-quadratic problem solved at each iteration must contain linear as well as quadratic terms in the objective function.

The parameters,  $B_0$  and  $b_0$ , in (5.82b) are computed by

$$B_0 = \nabla_{u_0} H_0(u_0^0) \quad (5.83a)$$

$$b_0 = H_0(u_0^0) - x_1^0 \quad (5.83b)$$

Similarly, the parameters,  $A_k$ ,  $B_k$  and  $b_k$ , in (5.82c) are computed by

$$A_k = \nabla_{x_k} H_k(x_k^0, u_k^0) \quad (5.84a)$$

$$B_k = \nabla_{u_k} H_k(x_k^0, u_k^0) \quad (5.84b)$$

$$b_k = H_k(x_k^0, u_k^0) - x_{k+1}^0 \quad (5.84c)$$

The Lagrangian of (5.82) is

$$\begin{aligned} L(\Delta x, \Delta u, \mu) = & \phi(\Delta x, \Delta u) - \mu'_0 (B'_0 \Delta u_0 - b_0) \\ & - \sum_{k=1}^{N-1} \mu'_k (A'_k \Delta x_k + B'_k \Delta u_k + b_k - \Delta x_{k+1}) \end{aligned} \quad (5.85)$$

However, for convenience in notation when illustrating the structure of the resulting KKT-system, we define  $\tilde{\mu}_k = -\mu_k$  such that the modified Lagrangian becomes

$$\begin{aligned} \tilde{L}(\Delta x, \Delta u, \tilde{\mu}) &= \phi(\Delta x, \Delta u) - \tilde{\mu}'_0(B'_0 \Delta u_0 - b_0) \\ &+ \sum_{k=1}^{N-1} \tilde{\mu}'_k(A'_k \Delta x_k + B'_k \Delta u_k + b_k - \Delta x_{k+1}) \end{aligned} \quad (5.86)$$

The necessary and sufficient optimality conditions for (5.82) in terms of the modified Lagrange multipliers are

$$R_0 \Delta u_0 + r_0 + B_0 \tilde{\mu}_0 = 0 \quad (5.87a)$$

$$\Delta x_1 = B'_0 \Delta u_0 + b_0 \quad (5.87b)$$

as well as

$$Q_k \Delta x_k + M_k \Delta u_k + q_k - \tilde{\mu}_{k-1} + A_k \tilde{\mu}_k = 0 \quad (5.87c)$$

$$M'_k \Delta x_k + R_k \Delta u_k + r_k + B_k \tilde{\mu}_k = 0 \quad (5.87d)$$

$$\Delta x_{k+1} = A'_k \Delta x_k + B'_k \Delta u_k + b_k \quad (5.87e)$$

and

$$P_N \Delta x_N + p_N - \tilde{\mu}_{N-1} = 0 \quad (5.87f)$$

In the case  $N = 3$ , the necessary and sufficient optimality conditions may be expressed as the KKT-system

$$\left[ \begin{array}{cccc|cccc} R_0 & & & & B_0 & & & \\ & Q_1 & M_1 & & -I & A_1 & & \\ & M'_1 & R_1 & & & B_1 & & \\ & & & Q_2 & M_2 & & & \\ & & & M'_2 & R_2 & & & \\ & & & & & & P_3 & \\ \hline B'_0 & -I & & & & & & \\ & A'_1 & B'_1 & -I & & & & \\ & & & A'_2 & B'_2 & -I & & \end{array} \right] \begin{bmatrix} \Delta u_0 \\ \Delta x_1 \\ \Delta u_1 \\ \Delta x_2 \\ \Delta u_2 \\ \Delta x_3 \\ \tilde{\mu}_0 \\ \tilde{\mu}_1 \\ \tilde{\mu}_2 \end{bmatrix} = - \begin{bmatrix} r_0 \\ q_1 \\ r_1 \\ q_2 \\ r_2 \\ p_3 \\ b_0 \\ b_1 \\ b_2 \end{bmatrix} \quad (5.88)$$

which may be rearranged to

$$\left[ \begin{array}{cccccccc|cccc} R_0 & B_0 & & & & & & & \Delta u_0 \\ B'_0 & 0 & -I & & & & & & \tilde{\mu}_0 \\ & & & -I & Q_1 & M_1 & A_1 & & \Delta x_1 \\ & & & & M'_1 & R_1 & B_1 & & \Delta u_1 \\ & & & & A'_1 & B'_1 & 0 & -I & \tilde{\mu}_1 \\ & & & & & & -I & Q_2 & M_2 & A_2 \\ & & & & & & & M'_2 & R_2 & B_2 \\ & & & & & & & A'_2 & B'_2 & 0 & -I \\ & & & & & & & & & -I & P_3 \end{array} \right] \begin{bmatrix} \Delta u_0 \\ \tilde{\mu}_0 \\ \Delta x_1 \\ \Delta u_1 \\ \tilde{\mu}_1 \\ \Delta x_2 \\ \Delta u_2 \\ \tilde{\mu}_2 \\ \Delta x_3 \end{bmatrix} = - \begin{bmatrix} r_0 \\ b_0 \\ q_1 \\ r_1 \\ b_1 \\ q_2 \\ r_2 \\ b_2 \\ p_3 \end{bmatrix} \quad (5.89)$$

The equivalent KKT systems (5.88) and (5.89) are examples of highly structured symmetric indefinite linear systems of equations. These systems easily

becomes very large for long time horizons. The following proposition contains a Riccati based solution procedure for (5.88) and (5.89). The recursion is obtained by a dynamic programming solution procedure of (5.82).

**Proposition 5.3.1 (Optimal Solution)**

The solution of (5.88) and (5.89) constitute the unique solution of (5.82) and may be obtained by the following procedure:

1) Compute the factorization

$$R_{e,k} = R_k + B_k P_{k+1} B'_k \quad (5.90a)$$

$$K_k = -R_{e,k}^{-1}(M'_k + B_k P_{k+1} A'_k) \quad (5.90b)$$

$$a_k = -R_{e,k}^{-1}(r_k + B_k(P_{k+1}b_k + p_{k+1})) \quad (5.90c)$$

$$P_k = Q_k + A_k P_{k+1} A'_k - K'_k R_{e,k} K_k \quad (5.90d)$$

$$p_k = q_k + A_k(P_{k+1}b_k + p_{k+1}) + K'_k(r_k + B_k(P_{k+1}b_k + p_{k+1})) \quad (5.90e)$$

for  $k = N-1, N-2, \dots, 1$  and

$$a_0 = -(R_0 + B_0 P_1 B'_0)^{-1}(r_0 + B_0(P_1 b_0 + p_1)) \quad (5.91)$$

2) Compute the primal solution  $\{\Delta u_k, \Delta x_{k+1}\}_{k=0}^{N-1}$  by

$$\Delta u_0 = a_0 \quad (5.92a)$$

$$\Delta x_1 = B'_0 \Delta u_0 + b_0 \quad (5.92b)$$

and

$$\Delta u_k = K_k \Delta x_k + a_k \quad (5.93a)$$

$$\Delta x_{k+1} = A'_k \Delta x_k + B'_k \Delta u_k + b_k \quad (5.93b)$$

for  $k = 1, 2, \dots, N-1$ .

3) Obtain the dual solution  $\{\tilde{\mu}_k\}_{k=0}^{N-1}$  by computing

$$\tilde{\mu}_{N-1} = P_N \Delta x_N + p_N \quad (5.94a)$$

$$\tilde{\mu}_{k-1} = A_k \tilde{\mu}_k + Q_k \Delta x_k + M_k \Delta u_k + q_k \quad (5.94b)$$

for  $k = N-1, N-2, \dots, 1$ .

*Proof.* See for instance Rao *et al.* (1998) or Ravn (1999). □

**Remark 5.3.2**

Uniqueness of the minimizer of (5.82) as well as existence of the inverses of  $R_{e,k} = R_k + B_k P_{k+1} B'_k$  in proposition 5.3.1 follows from the fact that  $R_0 = W_0$ ,  $\begin{bmatrix} Q_k & M_k \\ M'_k & R_k \end{bmatrix} = W_k$ , and  $P_N = W_N$  are all symmetric positive definite matrices by construction.

**Remark 5.3.3**

To retain numerical stability of the Riccati based recursion, symmetry of the sequence matrices  $\{P_k\}_{k=0}^{N-1}$  must be explicitly enforced. If symmetry is not enforced the recursion may diverge for ill-conditioned systems. Another stable alternative implementation of the Riccati recursion is based on orthogonal factorizations as described by Bierman (1977) and Kailath *et al.* (2000).

In the sequential quadratic programming algorithm as outlined,  $\mu_k$  rather than  $\tilde{\mu}_k$  is needed. The recursion for  $\mu_k$  is stated in the following corollary.

**Corollary 5.3.4**

Let  $\mu_k = -\tilde{\mu}_k$  for  $k = 0, 1, \dots, N-1$ . Then  $\{\mu_k\}_{k=0}^{N-1}$  may be obtained by

$$\mu_{N-1} = -P_N \Delta x_N - p_N \quad (5.95a)$$

$$\mu_{k-1} = A_k \mu_k - Q_k \Delta x_k - M_k \Delta u_k - q_k \quad (5.95b)$$

*Proof.* Follows directly from the fact that  $\mu_k = -\tilde{\mu}_k$ . □

Let  $x_k \in \mathbb{R}^n$  and  $u_k \in \mathbb{R}^m$ . Then the Riccati based method for solution of (5.82) has computational complexity  $O(N(n^3 + n^2m))$  while direct solution of either (5.88) or (5.89) by dense symmetric indefinite factorization has complexity  $O(N^3(n+m)^3)$ . In particular for long horizons,  $N$ , the Riccati based factorization is much more efficient than the dense direct solution procedure. An alternative solution procedure is to eliminate the states in (5.82) and obtain a mathematical program with  $u$  as the decision variables. Solution of the resulting KKT system has complexity  $O(N^3m^3)$  but the formation of the matrices describing the reduced system is in itself an  $O(N(n^2m))$  process. For systems with very large state dimension memory considerations and the need to or not to store all sensitivities along the current trajectory may determine the applicability of an algorithm. The Riccati based factorization requires storage of all sensitivities while alternative dense feasible path algorithms do not necessarily have such a requirement.

Convergence in the SQP algorithm is checked just after computation of the search direction. The convergence criteria (5.22) specialized to (5.64) becomes

$$\begin{aligned} & |r'_0 \Delta u_0 + \left( \sum_{k=1}^{N-1} q'_k \Delta x_k + r'_k \Delta u_k \right) + p'_N \Delta x_N| \\ & + \sum_{k=0}^{N-1} \sum_{i=1}^n |[\mu_k]_i [b_k]_i| \leq \varepsilon \end{aligned} \quad (5.96)$$

### 5.3.6 Linesearch and Penalty Functions

Using the step length computed by proposition 5.3.1 and corollary 5.3.4, the new iterate,  $(x, u, \pi)$ , in the SQP algorithm is generated by

$$\begin{bmatrix} x \\ u \\ \pi \end{bmatrix} = \begin{bmatrix} x^0 \\ u^0 \\ \pi^0 \end{bmatrix} + \alpha \begin{bmatrix} \Delta x \\ \Delta u \\ \mu - \pi^0 \end{bmatrix} \quad (5.97)$$

in which  $\alpha$  is the step length. Let

$$g_0(u_0, x_1) = H_0(u_0) - x_1 \quad (5.98a)$$

$$g_k(x_k, u_k, x_{k+1}) = H_k(x_k, u_k) - x_{k+1}, \quad k \in \mathcal{N} \quad (5.98b)$$

such that the constraints (5.64b) and (5.64c) may be expressed as

$$g_0(u_0, x_1) = 0 \quad (5.99a)$$

$$g_k(x_k, u_k, x_{k+1}) = 0 \quad (5.99b)$$

Powell's  $l_1$ -penalty function (5.36) for the discrete-time nonlinear optimal control problem (5.64) may be expressed as

$$\begin{aligned} P(x, u, \sigma) = & f(x, u) + \sum_{i=1}^n \sigma_{i,0} |[g_0(u_0, x_1)]_i| \\ & + \sum_{k=1}^{N-1} \sum_{i=1}^n \sigma_{i,k} |[g_k(x_k, u_k, x_{k+1})]_i| \end{aligned} \quad (5.100)$$

in which  $f(x, u)$  is given by (5.64a). Except at the first iteration, the constraint violation penalty parameters are computed by

$$\sigma_{i,k} \leftarrow \max \left\{ |[\mu_k]_i|, \frac{1}{2}(\sigma_{i,k} + |[\mu_k]_i|) \right\} \quad (5.101)$$

for  $i = 1, \dots, n$  and  $k = 0, 1, \dots, N-1$ . At the first iteration  $\sigma_{i,k} = |\mu_{i,k}|$ . The merit function used for selecting the step length  $\alpha$  is

$$T(\alpha) = P(x^0 + \alpha \Delta x, u^0 + \alpha \Delta u, \sigma) \quad (5.102)$$

and its computation in an inexact line search procedure is facilitated by the following lemma.

**Lemma 5.3.5**

Let  $(\Delta x, \Delta u)$  be a solution of (5.82) and let  $T : [0, 1] \mapsto \mathbb{R}$  be defined by (5.102) and (5.100). Let  $f(x, u)$  be defined by (5.64a). Let  $g_0$  as well as  $g_k$  for  $k \in \mathcal{N}$  be defined by (5.98). Furthermore, let

$$\tilde{b}_0(\alpha) = g_0(u_0^0 + \alpha \Delta u, x_1^0 + \alpha \Delta x_1) \quad (5.103a)$$

$$\tilde{b}_k(\alpha) = g_k(x_k^0 + \alpha \Delta x_k, u_k^0 + \alpha \Delta u_k, x_{k+1}^0 + \alpha \Delta x_{k+1}) \quad (5.103b)$$

for  $k \in \mathcal{N}$ . Then

$$T(\alpha) = f(x^0 + \alpha \Delta x, u^0 + \alpha \Delta u) + \sum_{k=0}^{N-1} \sum_{i=1}^n \sigma_{i,k} |\tilde{b}_k(\alpha)|_i \quad (5.104)$$

and

$$T(0) = f(x^0, u^0) + \sum_{k=0}^{N-1} \sum_{i=1}^n \sigma_{i,k} |[b_k]_i| \quad (5.105a)$$

$$T'(0) = r'_0 \Delta u_0 + \left( \sum_{k=1}^{N-1} q'_k \Delta x_k + r'_k \Delta u_k \right) + p'_N \Delta x_N - \sum_{k=0}^{N-1} \sum_{i=1}^n \sigma_{i,k} |[b_k]_i| \quad (5.105b)$$

in which  $b_k$ ,  $r_k$ ,  $q_k$  and  $p_N$  are defined by (5.81).  $b_0$  is defined by (5.83b) and  $b_k$  is defined by (5.84c).

*Proof.* The results are direct specializations of the results in lemma 5.2.8 to the discrete-time nonlinear optimal control problem (5.64).  $\square$

It is only the computation of the line search merit function,  $T(\alpha)$ , that is different from the standard SQP algorithm when solving the discrete-time nonlinear optimal control problem (5.64). The remaining parts of the line search algorithm are identical to the steps stated in algorithm 7.

**Algorithm 8 (Line Search for (5.64))**

Let  $f^0 = f(x^0, u^0)$ ,  $\Delta f = r'_0 \Delta u_0 + \left( \sum_{k=1}^{N-1} q'_k \Delta x_k + r'_k \Delta u_k \right) + p'_N \Delta x_N$ ,  $b_k$ ,  $x^0$ ,  $u^0$ ,  $\Delta x$ ,  $\Delta u$ , and  $\sigma$  be given.

1. Compute  $\eta_0 = \sum_{k=0}^{N-1} \sum_{i=1}^n \sigma_{i,k} |[b_k]_i|$ ,  $T(0) = f^0 + \eta_0$ , and  $T'(0) = \Delta f - \eta_0$ . Set  $\alpha_1 = 1$ .
2. Compute  $x = \{x_k = x_k^0 + \Delta x_k\}_{k=1}^N$  and  $u = \{u_k = u_k^0 + \Delta u_k\}_{k=0}^{N-1}$ . Evaluate  $f = f(x, u)$  by (5.64a). Evaluate  $g_0 = g_0(u_0, x_1)$  and  $g_k = g_k(x_k, u_k, x_{k+1})$  by (5.98). Compute  $T(\alpha_1) = f + \sum_{k=0}^{N-1} \sum_{i=1}^n \sigma_{i,k} |[g_k]_i|$ .
3. If  $T(\alpha_1) \leq T(0) + c_1 T'(0)$  then stop with  $\alpha = \alpha_1$ .
4. Compute  $\alpha_{\min}$  by (5.48) and  $\alpha_2$  by (5.49).
5. Compute  $x = \{x_k = x_k^0 + \alpha_2 \Delta x_k\}_{k=1}^N$  and  $u = \{u_k = u_k^0 + \alpha_2 \Delta u_k\}_{k=0}^{N-1}$ . Evaluate  $f = f(x, u)$  by (5.64a). Evaluate  $g_0 = g_0(u_0, x_1)$  and  $g_k = g_k(x_k, u_k, x_{k+1})$  by (5.98). Compute  $T(\alpha_2) = f + \sum_{k=0}^{N-1} \sum_{i=1}^n \sigma_{i,k} |[g_k]_i|$ .
6. If  $T(\alpha_2) \leq T(0) + c_1 \alpha_2 T'(0)$  then stop with  $\alpha = \alpha_2$ .
7. Compute  $a$  and  $b$  by (5.51). If  $a = 0$  compute  $\alpha_{\min} = \frac{-T'(0)}{b}$ . Otherwise compute  $\alpha_{\min}$  by (5.52). Compute  $\alpha$  by (5.53).



8. Compute  $x = \{x_k = x_k^0 + \alpha \Delta x_k\}_{k=1}^N$  and  $u = \{u_k = u_k^0 + \alpha \Delta u_k\}_{k=0}^{N-1}$ . Evaluate  $f = f(x, u)$  by (5.64a). Evaluate  $g_0 = g_0(u_0, x_1)$  and  $g_k = g_k(x_k, u_k, x_{k+1})$  by (5.98). Compute  $T(\alpha) = f + \sum_{k=0}^{N-1} \sum_{i=1}^n \sigma_{i,k} |[g_k]_i|$ .
9. If  $T(\alpha) \leq T(0) + c_1 \alpha T'(0)$  then stop.
10. Set  $\alpha_1 = \alpha_2$ ,  $\alpha_2 = \alpha$ ,  $T(\alpha_1) = T(\alpha_2)$ , and  $T(\alpha_2) = T(\alpha)$ . Go to step (7).

The algorithm returns  $\alpha$ ,  $(x, u) = \{x_{k+1}, u_k\}_{k=0}^{N-1}$ ,  $f$ , and  $\{g_k\}_{k=0}^{N-1}$ .

### 5.3.7 The SQP Algorithm for Optimal Control

To accommodate solution of the nonlinear optimal control problem (5.64) with an SQP algorithm similar to the SQP algorithm used for optimization of equality constrained nonlinear programs (5.1), a number of specializations have been introduced. The evaluation of the nominal trajectory and the gradients has been tailored to the optimal control problem. Based on these gradients a block partitioned, structured update of the Hessian approximation specialized to the optimal control problem has been introduced. The structure of this approximate Hessian matrix and the staircase structure of the dynamic constraints enable development of an efficient Riccati based solution procedure for the quadratic program arising at each iteration in the SQP algorithm for the nonlinear optimal control problem. Further, the line search algorithm has been tailored to the nonlinear optimal control problem. Substitution of these specializations to the nonlinear optimal control problem in the SQP algorithm (algorithm 6) gives an SQP algorithm for solution of the nonlinear optimal control problem (5.64).

#### Algorithm 9 (SQP for Optimal Control (5.64))

Let  $\{x_{k+1}^0, u_k^0, \pi_k^0\}_{k=0}^{N-1}$  be given.

1. Evaluate the functions and gradient at the initial point: Calculate  $f^0 = f(x^0, u^0)$  by (5.64a),  $b_0 = g_0^0 = g_0(u_0^0, x_1^0)$  and  $b_k = g_k^0 = g_k(x_k^0, u_k^0, x_{k+1}^0)$  by (5.98),  $(r_0, \{q_k, r_k\}_{k=1}^{N-1}, p_N)$  by (5.81), and  $(B_0, \{A_k, B_k\}_{k=1}^{N-1})$  by (5.83) and (5.84).
2. Choose initial approximations for the Hessian block matrices:  $\{W_k\}_{k=0}^N$  are chosen as some symmetric positive definite matrices. These matrices constitute an initial approximation of the Hessian matrix. Define  $R_0, \{Q_k, M_k, R_k\}_{k=1}^{N-1}$  and  $P_N$  according to (5.79).
3. Compute  $(\{\Delta x_{k+1}, \Delta u_k, \mu_k\}_{k=0}^{N-1})$  by proposition 5.3.1 and corollary 5.3.4 using the data  $(R_0, r_0, B_0, b_0, \{Q_k, M_k, R_k, q_k, r_k, A_k, B_k, b_k\}_{k=1}^{N-1}, P_N, p_N)$ .
4. Compute  $\Delta f = r_0' \Delta u_0 + \sum_{k=1}^{N-1} q_k' \Delta x_k + r_N' \Delta u_k + p_N' \Delta x_N$  and  $\eta = \sum_{k=0}^{N-1} \sum_{i=1}^n |[\mu_k]_i [b_k]_i|$ . If  $|\Delta f| + \eta \leq \epsilon$  then terminate with  $\{x_{k+1}^0, u_k^0, \mu_k\}_{k=0}^{N-1}$  as the solution. Return  $f^0$  as well.
5. Compute  $\alpha$  by the line search algorithm using Powell's  $l_1$ -merit function (algorithm 8). The result is  $\alpha$ ,  $\{x_{k+1}, u_k\}_{k=0}^{N-1}$ ,  $f = f(x, u)$ , and  $(g_0 = g_0(u_0, x_1), \{g_k = g_k(x_k, u_k, x_{k+1})\}_{k=1}^{N-1})$ .
6. Compute  $\{\pi_k = \pi_k^0 + \alpha(\mu_k - \pi_k^0)\}_{k=0}^{N-1}$ .

7. Compute

$$\nabla_{u_0} \mathcal{L}^0 = r_0 - B_0 \pi_0 \quad (5.106a)$$

$$\nabla_{x_k} \mathcal{L}^0 = q_k - A_k \pi_k + \pi_{k-1} \quad k \in \mathcal{N} \quad (5.106b)$$

$$\nabla_{u_k} \mathcal{L}^0 = r_k - B_k \pi_k \quad k \in \mathcal{N} \quad (5.106c)$$

$$\nabla_{x_N} \mathcal{L}^0 = p_N + \pi_{N-1} \quad (5.106d)$$

8. Evaluate the gradients at the new point, i.e. compute

$$r_0 = \nabla_{u_0} f_0(u_0) \quad (5.107a)$$

$$q_k = \nabla_{x_k} f_k(x_k, u_k) \quad k \in \mathcal{N} \quad (5.107b)$$

$$r_k = \nabla_{u_k} f_k(x_k, u_k) \quad k \in \mathcal{N} \quad (5.107c)$$

$$p_N = \nabla_{x_N} f_N(x_N) \quad (5.107d)$$

and

$$B_0 = \nabla_{u_0} H_0(u_0) \quad (5.108a)$$

$$A_k = \nabla_{x_k} H_k(x_k, u_k) \quad k \in \mathcal{N} \quad (5.108b)$$

$$B_k = \nabla_{u_k} H_k(x_k, u_k) \quad k \in \mathcal{N} \quad (5.108c)$$

9. Compute

$$\nabla_{u_0} \mathcal{L} = r_0 - B_0 \pi_0 \quad (5.109a)$$

$$\nabla_{x_k} \mathcal{L} = q_k - A_k \pi_k + \pi_{k-1} \quad k \in \mathcal{N} \quad (5.109b)$$

$$\nabla_{u_k} \mathcal{L} = r_k - B_k \pi_k \quad k \in \mathcal{N} \quad (5.109c)$$

$$\nabla_{x_N} \mathcal{L} = p_N + \pi_{N-1} \quad (5.109d)$$

10. Compute

$$s_0 = u_0 - u_0^0 \quad (5.110a)$$

$$y_0 = \nabla_{u_0} \mathcal{L} - \nabla_{u_0} \mathcal{L}^0 \quad (5.110b)$$

and

$$s_k = \begin{bmatrix} x_k - x_k^0 \\ u_k - u_k^0 \end{bmatrix} \quad k \in \mathcal{N} \quad (5.111a)$$

$$y_k = \begin{bmatrix} \nabla_{x_k} \mathcal{L} - \nabla_{x_k} \mathcal{L}^0 \\ \nabla_{u_k} \mathcal{L} - \nabla_{u_k} \mathcal{L}^0 \end{bmatrix} \quad k \in \mathcal{N} \quad (5.111b)$$

as well as

$$s_N = x_N - x_N^0 \quad (5.112a)$$

$$y_N = \nabla_{x_N} \mathcal{L} - \nabla_{x_N} \mathcal{L}^0 \quad (5.112b)$$

11. For  $k = 0, 1, \dots, N$  compute

$$\theta_k = \begin{cases} 1 & s'_k y_k \geq 0.2 s'_k W_k s_k \\ \frac{0.8 s'_k W_k s_k}{s'_k W_k s_k - s'_k y_k} & s'_k y_k < 0.2 s'_k W_k s_k \end{cases} \quad (5.113a)$$

$$r_k = \theta_k y_k + (1 - \theta_k) W_k s_k \quad (5.113b)$$

$$W_k \leftarrow \begin{cases} W_k & \|s_k\|_2 \leq \varepsilon_s \\ W_k - \frac{W_k s_k s'_k W_k}{s'_k W_k s_k} + \frac{r_k r'_k}{s'_k r_k} & \|s_k\|_2 > \varepsilon_s \end{cases} \quad (5.113c)$$

Define  $R_0, \{Q_k, M_k, R_k\}_{k=1}^{N-1}$ , and  $P_N$  according to (5.79).

12. Update the old values:  $\{x_{k+1}^0, u_k^0, \pi_k^0\}_{k=0}^{N-1} \leftarrow \{x_{k+1}, u_k, \pi_k\}_{k=0}^{N-1}$ ,  $f^0 \leftarrow f$ , and  $\{b_k\}_{k=0}^{N-1} \leftarrow \{g_k\}_{k=0}^{N-1}$ . Go to (3).

Algorithm 9 is an infeasible path method. As a consequence the integration and optimization is performed simultaneously. This implies that the algorithm can be applied to unstable systems without stability problems. However, the constraints are only guaranteed to be satisfied at the solution but not at intermediate iterates.

### 5.3.8 Discretization

The remaining issue in having a complete procedure to compute an approximate local minimizer of (5.54) by computation of a local minimizer of (5.64) concerns the time discretization of the continuous time optimal control problem (5.54). As already emphasized for *illustrative* reasons, the explicit Euler method with a fixed step length is used to discretize the continuous time nonlinear optimal control problem. Proposition 5.3.6 provides the equations constituting an explicit Euler integration procedure with sensitivities.

#### Proposition 5.3.6 (Explicit Euler Discretization)

Consider the continuous time nonlinear optimal control problem (5.54) and the discrete-time nonlinear optimal control problem (5.64). Let the considered discrete times  $t_k$  be equidistant such that

$$t_k = t_0 + \tau k \quad k = 0, 1, \dots, N \quad \tau = \frac{t_f - t_0}{N} \quad (5.114)$$

in which  $\tau$  is the constant step length.  $x_k = x(t_k)$  and  $u_k = u(t_k)$ .

Then the explicit Euler approximation of the objective function (5.54a) is

$$f(x, u) = f_0(u_0) + \sum_{k=1}^{N-1} f_k(x_k, u_k) + f_N(x_N) \quad (5.115)$$

in which

$$f_0(u_0) = \tau l(x_0, u_0, t_0) \quad (5.116a)$$

$$f_k(x_k, u_k) = \tau l(x_k, u_k, t_k) \quad k \in \mathcal{N} \quad (5.116b)$$

$$f_N(x_N) = L(x_N, t_N) \quad (5.116c)$$

The gradients of the explicit Euler approximation of the objective function are

$$r_0 = \nabla_{u_0} f_0(u_0) = \tau \nabla_u l(x_0, u_0, t_0) \quad (5.117a)$$

$$q_k = \nabla_{x_k} f_k(x_k, u_k) = \tau \nabla_x l(x_k, u_k, t_k) \quad (5.117b)$$

$$r_k = \nabla_{u_k} f_k(x_k, u_k) = \tau \nabla_u l(x_k, u_k, t_k) \quad (5.117c)$$

$$p_N = \nabla_{x_N} f_N(x_N) = \nabla_x L(x_N, t_N) \quad (5.117d)$$

in which  $k \in \mathcal{N}$ .

The first time step is special as  $x_0$  is a parameter. At the first time step, the explicit Euler approximation of (5.54b) is

$$x_1 = H_0(u_0) \quad (5.118)$$

in which the function  $H_0(u_0)$  and its derived quantities are

$$H_0(u_0) = x_0 + \tau h(x_0, u_0, t_0) \quad (5.119a)$$

$$B_0 = \nabla_{u_0} H_0(u_0) = \tau \nabla_u h(x_0, u_0, t_0) \quad (5.119b)$$

$$b_0 = g_0(u_0, x_1) = H_0(u_0) - x_1 \quad (5.119c)$$

As subsequent time steps,  $k \in \mathcal{N}$ , the explicit Euler approximation of (5.54b) is

$$x_{k+1} = H_k(x_k, u_k) \quad (5.120)$$

in which the function  $H_k(x_k, u_k)$  and its derived quantities are

$$H_k(x_k, u_k) = x_k + \tau h(x_k, u_k, t_k) \quad (5.121a)$$

$$\begin{aligned} A_k &= \nabla_{x_k} H_k(x_k, u_k) \\ &= I + \tau \nabla_x h(x_k, u_k, t_k) \end{aligned} \quad (5.121b)$$

$$\begin{aligned} B_k &= \nabla_{u_k} H_k(x_k, u_k) \\ &= \tau \nabla_u h(x_k, u_k, t_k) \end{aligned} \quad (5.121c)$$

$$\begin{aligned} b_k &= g_k(x_k, u_k, x_{k+1}) \\ &= H_k(x_k, u_k) - x_{k+1} \end{aligned} \quad (5.121d)$$

*Proof.* The results follow straightforwardly from the fact that the explicit equidistant Euler discretization of  $\dot{x}(t) = f(x(t))$  is obtained by  $t_k = t_0 + \tau k$  and  $x_{k+1} = x_k + \tau f(x_k)$ .  $\square$

## 5.4 Feasible Path SQP

The model equations (5.64b)-(5.64c) are only guaranteed to be satisfied at the optimal solution in the *infeasible* path sequential quadratic programming algorithm. An alternative to this algorithm is the *feasible* path sequential quadratic programming algorithm. By this method, the constraints (5.64b)-(5.64c) of (5.64) are satisfied at every iteration and not only at the optimal solution. As already explained, the feasible path SQP algorithm for solution of

the unconstrained optimal control problem may be regarded as a control vector parameterization (CVP) method.

Use of a feasible path SQP algorithm for the dynamic optimization in model predictive control allows early termination of the algorithm due to the real-time constraint and application of the latest non-optimal iterate to the plant (c.f. Scokaert *et al.*, 1999; Tenny *et al.*, 2002). Scokaert *et al.* (1999) demonstrate that such a suboptimal procedure implies stability provided that the solution is feasible. The major problem with the feasible path SQP algorithm concerns its application to unstable processes. In such cases, an open-loop implementation of the algorithm may terminate due to instability of the process and the possibility of finite escape time for nonlinear systems. In a closed-loop implementation of the feasible path SQP algorithm, the problems concerned with unstable processes are overcome.

The major difference between the infeasible path SQP algorithm and the feasible path SQP algorithm is the functions used in the line search procedure and the forward solution of the model. The infeasible path SQP algorithm uses a *merit* function, i.e. Powell's  $l_1$  merit function, and computes a new iterate by  $x_k = x_k^0 + \alpha \Delta x_k$  and  $u_k = u_k^0 + \alpha \Delta u_k$ . In contrast, the open-loop implementation of the feasible path SQP algorithm computes a new iterate by

$$u_k = u_k^0 + \alpha \Delta u_k \quad (5.122a)$$

$$x_{k+1} = H_k(x_k, u_k) \quad x_0 = x_0^0 \quad (5.122b)$$

and adjusts the step length,  $\alpha$ , using the objective function (5.64a) as its merit function. It is obvious by this construction that the constraints are always satisfied. The problem is, that the open-loop control sequence  $\{u_k\}_{k=0}^{N-1}$  does not necessarily stabilize the process  $x_{k+1} = H_k(x_k, u_k)$ . Numerically, such instability phenomena manifest themselves by divergence and termination of the algorithm. Precautions can be taken such that the objective function and the norm of the states are monitored along the integration of the model, i.e. solution of (5.122b). If these monitored variables become excessively large, the integration is stopped and the step length reduced. Such a procedure requires that the initial control profile  $\{u_k^0\}_{k=0}^{N-1}$  does not lead to divergence. An alternative procedure proposed by Tenny *et al.* (2002) extend the pre-stabilization methodology of Rossitier *et al.* (1998) to nonlinear systems. This alternative procedure is the closed-loop feasible path SQP algorithm. Compared to the open-loop feasible path controls,  $u_k^0 + \alpha \Delta u_k$ , the controls,  $u_k$ , computed in the closed-loop feasible path SQP algorithm are extended with a term corresponding to proportional feedback from the prediction error,  $x_k - (x_k^0 + \alpha \Delta x_k)$ , of the linear model. The feedback term reduces the sensitivity and stabilizes the current trajectory numerically. Hence, the new iterate in the closed-loop feasible path SQP algorithm is computed by

$$u_k = u_k^0 + \alpha \Delta u_k + K_k(x_k - [x_k^0 + \alpha \Delta x_k]) \quad (5.123a)$$

$$x_{k+1} = H_k(x_k, u_k) \quad (5.123b)$$

in which  $K_k$  is some stabilizing feedback gain matrix. The merit of this iterate and selection of the step length,  $\alpha$ , is based on the objective function as all the constraints are feasible.

As stated above, the search direction needed in the open-loop feasible path SQP algorithm is  $\{\Delta u_k\}_{k=0}^{N-1}$ , while the closed-loop feasible path SQP algorithm uses  $\{\Delta x_{k+1}, \Delta u_k\}_{k=0}^{N-1}$  as search direction. In addition, the closed loop feasible path SQP algorithm needs a sequence of stabilizing feedback gains,  $\{K_k\}_{k=1}^{N-1}$ . The search direction is found by solving the quadratic program (5.82). One method to solve this program is by application of proposition 5.3.1 and corollary 5.3.4. However,  $\{b_k = H_k(x_k, u_k) - x_{k+1} = 0\}_{k=0}^{N-1}$  in the feasible path SQP algorithms. This implies that the recursions in proposition 5.3.1 may be simplified. The result of this simplification and hence a procedure for computing the search direction is stated in the following proposition.

**Proposition 5.4.1**

Consider the case when  $b_k = 0$  for  $k = 0, 1, \dots, N-1$ . Let (5.82) be a convex quadratic program. Then the solution of (5.82) may be obtained by the following procedure:

1) Compute the factorization

$$R_{e,k} = R_k + B_k P_{k+1} B'_k \quad (5.124a)$$

$$K_k = -R_{e,k}^{-1} (M'_k + B_k P_{k+1} A'_k) \quad (5.124b)$$

$$a_k = -R_{e,k}^{-1} (r_k + B_k p_{k+1}) \quad (5.124c)$$

$$P_k = Q_k + A_k P_{k+1} A'_k - K'_k R_{e,k} K_k \quad (5.124d)$$

$$p_k = q_k + A_k p_{k+1} + K'_k (r_k + B_k p_{k+1}) \quad (5.124e)$$

for  $k = N-1, N-2, \dots, 1$  and

$$a_0 = -(R_0 + B_0 P_1 B'_0)^{-1} (r_0 + B_0 p_1) \quad (5.125)$$

2) Compute the primal solution  $\{\Delta u_k, \Delta x_{k+1}\}_{k=0}^{N-1}$  by

$$\Delta u_0 = a_0 \quad (5.126a)$$

$$\Delta x_1 = B'_0 \Delta u_0 \quad (5.126b)$$

and

$$\Delta u_k = K_k \Delta x_k + a_k \quad (5.127a)$$

$$\Delta x_{k+1} = A'_k \Delta x_k + B'_k \Delta u_k \quad (5.127b)$$

for  $k = 1, 2, \dots, N-1$ .

3) Obtain the dual solution  $\{\mu_k\}_{k=0}^{N-1}$  by the recursion

$$\mu_{N-1} = -P_N \Delta x_N - p_N \quad (5.128a)$$

$$\mu_{k-1} = A_k \mu_k - Q_k \Delta x_k - M_k \Delta u_k - q_k \quad (5.128b)$$

for  $k = N-1, N-2, \dots, 1$ .

*Proof.* Follows directly from proposition 5.3.1 and corollary 5.3.4 when  $\{b_k = 0\}_{k=0}^{N-1}$  is inserted.  $\square$

### 5.4.1 Open-Loop Feasible Path SQP

The step length,  $\alpha$ , in the open-loop feasible path SQP algorithm for solution of (5.82) is based on the Armijo condition

$$T(\alpha) \leq T(0) + c_1 \alpha T'(0) \quad (5.129)$$

in which the merit function,  $T(\alpha)$ , is the objective function (5.64a) along the current search direction

$$T(\alpha) = \sum_{k=0}^{N-1} f_k(x_k(\alpha), u_k(\alpha)) + f_N(x_N(\alpha)) \quad (5.130)$$

and  $f_0(u_0(\alpha)) = f_0(x_0(\alpha), u_0(\alpha))$ . In the open-loop feasible path SQP algorithm, the inputs as function of the step length,  $u_k = u_k(\alpha)$ , are

$$u_k(\alpha) = u_k^0 + \alpha \Delta u_k \quad k = 0, 1, \dots, N-1 \quad (5.131)$$

Similarly, the states as function of the step length,  $x_k = x_k(\alpha)$ , are computed by

$$x_0(\alpha) = x_0 \quad (5.132a)$$

$$x_{k+1}(\alpha) = H_k(x_k(\alpha), u_k(\alpha)) \quad (5.132b)$$

for  $k = 0, \dots, N-1$ . By these definitions  $T(0)$  and  $T'(0)$  used in the Armijo condition may be computed according to the following lemma.

**Lemma 5.4.2**

Let  $\{\Delta x_{k+1}, \Delta u_k\}_{k=0}^{N-1}$  be the solution of (5.82). Let  $T : [0, 1] \mapsto \mathbb{R}$  be defined by (5.130). Let  $\{u_k(\alpha)\}_{k=0}^{N-1}$  be defined by (5.131) and let  $\{x_k(\alpha)\}_{k=0}^N$  be defined by (5.132). Let  $f^0$  be the value of (5.64a) at the nominal trajectory  $(\{x_{k+1}^0, u_k^0\}_{k=0}^{N-1})$ . Let  $(r_0, \{q_k, r_k\}_{k=1}^{N-1}, p_N)$  be defined by (5.81) at the nominal trajectory  $\{x_{k+1}^0, u_k^0\}_{k=0}^{N-1}$ . Let  $(B_0, \{A_k, B_k\}_{k=1}^{N-1})$  be defined by (5.83a) and (5.84a)-(5.84b), respectively, at the nominal trajectory  $\{x_{k+1}^0, u_k^0\}_{k=0}^{N-1}$ .

Then

$$T(0) = \sum_{k=0}^{N-1} f_k(x_k^0, u_k^0) + f_N(x_N^0) = f^0 \quad (5.133)$$

and

$$T'(0) = \sum_{k=0}^{N-1} r'_k \Delta u_k + \sum_{k=1}^{N-1} q'_k \frac{dx_k}{d\alpha}(0) + p'_N \frac{dx_N}{d\alpha}(0) \quad (5.134)$$

in which  $\frac{dx_k}{d\alpha}(0)$  is obtained by the recursion

$$\frac{dx_0}{d\alpha}(0) = 0 \quad (5.135a)$$

$$\frac{dx_{k+1}}{d\alpha}(0) = A'_k \frac{dx_k}{d\alpha}(0) + B'_k \Delta u_k \quad (5.135b)$$

*Proof.* (5.133) follows immediately by the fact that  $x_k(0) = x_k^0$  and  $u_k(0) = u_k^0$ . (5.134) is obtained by application of the chain rule of differentiation.  $\square$

With the modification introduced above, the solution of the nonlinear optimal control problem (5.64) by the open-loop feasible path SQP method consists of the steps stated in algorithm 10 and algorithm 11.

**Algorithm 10 (Feasible Path SQP for (5.64))**

Let  $x_0^0 = x_0$  and  $\{u_k^0\}_{k=0}^{N-1}$  be given.

1. Compute the initial state profile and objective function:  $\{x_{k+1}^0 = H_k(x_k^0, u_k^0)\}_{k=0}^{N-1}$  and  $f^0 = \sum_{k=0}^{N-1} f_k(x_k^0, u_k^0) + f_N(x_N^0)$ .
2. Evaluate the gradients along the nominal trajectory: Compute  $(r_0, \{q_k, r_k\}_{k=1}^{N-1}, p_N)$  by (5.81) and  $(B_0, \{A_k, B_k\}_{k=1}^{N-1})$  by (5.83a) and (5.84a)-(5.84b).
3. Compute an initial estimate of the Lagrange multipliers:  $\pi_{N-1}^0 = p_N$ , and  $\{\pi_{k-1}^0 = A_k \pi_k^0 - q_k\}_{k=N-1}^1$ .
4. Choose initial approximations for the Hessian block matrices:  $\{W_k\}_{k=0}^N$  are chosen as some symmetric positive definite matrices. Define  $R_0, \{Q_k, M_k, R_k\}_{k=1}^{N-1}$  and  $P_N$  according to (5.79).
5. Compute  $(\{\Delta x_{k+1}, \Delta u_k, \mu_k\}_{k=0}^{N-1})$  by proposition 5.4.1 using the data  $(R_0, r_0, B_0, \{Q_k, M_k, R_k, q_k, r_k, A_k, B_k\}_{k=1}^{N-1}, P_N, p_N)$ .
6. Check for convergence: Compute  $\Delta f = r_0' \Delta u_0 + \sum_{k=1}^{N-1} q_k' \Delta x_k + r_k' \Delta u_k + p_N' \Delta x_N$ . If  $|\Delta f| \leq \epsilon$  then terminate with  $\{x_{k+1}^0, u_k^0, \mu_k\}_{k=0}^{N-1}$  as the solution. Return  $f^0$  as well.
7. Compute  $\alpha$  by the line search algorithm (algorithm 11). The result is  $\{x_{k+1}, u_k\}_{k=0}^{N-1}$  and  $f = f(x, u)$ .
8. Compute  $\{\pi_k = \pi_k^0 + \alpha(\mu_k - \pi_k^0)\}_{k=0}^{N-1}$ .
9. Compute the Lagrange derivatives using  $\{x_{k+1}^0, u_k^0, \pi_k\}_{k=0}^{N-1}$  (see step 7 of algorithm 9).
10. Evaluate the gradients along the trajectory  $\{x_{k+1}, u_k\}_{k=0}^{N-1}$ : Compute  $(r_0, \{q_k, r_k\}_{k=1}^{N-1}, p_N)$  by (5.81) and  $(B_0, \{A_k, B_k\}_{k=1}^{N-1})$  by (5.83a) and (5.84a)-(5.84b).
11. Compute the Lagrange derivatives using  $\{x_{k+1}, u_k, \pi_k\}_{k=0}^{N-1}$  (see step 9 of algorithm 9).
12. Compute  $\{s_k, y_k\}_{k=0}^N$  as in step 10 of algorithm 9.
13. Update the Hessian approximation as in step 11 of algorithm 9.
14. Update the old values:  $\{x_{k+1}^0, u_k^0, \pi_k^0\}_{k=0}^{N-1} \leftarrow \{x_{k+1}, u_k, \pi_k\}_{k=0}^{N-1}$  and  $f^0 \leftarrow f$ . Go to (5).

The computation of the initial approximation of the Lagrange multipliers,  $\{\pi_k^0\}_{k=0}^{N-1}$ , in step (3) is based on (5.69b) and (5.69d). The operations of the line search algorithm adapted for the open-loop feasible path SQP algorithm, i.e. algorithm 10, are stated in algorithm 11.



**Algorithm 11 (Line Search for FSQP of (5.64))**

Let  $x^0 = \{x_k^0\}_{k=1}^N$  and  $u^0 = \{u_k^0\}_{k=0}^{N-1}$ . Let  $f^0 = f(x^0, u^0)$ ,  $\eta = \sum_{k=0}^{N-1} r'_k \Delta u_k$ ,  $\{u_k^0\}_{k=0}^{N-1}$ ,  $\{\Delta u_k\}_{k=0}^{N-1}$ ,  $(\{q_k\}_{k=1}^{N-1}, p_N)$ , and  $(B_0, \{A_k, B_k\}_{k=1}^{N-1})$  be given. Let  $c_1 = 10^{-4}$ .

1. Set  $T(0) = f^0$  and  $x'_0(0) = 0$ . Compute  $x'_{k+1}(0) = A'_k x'_k(0) + B'_k \Delta u_k$  for  $k = 0, 1, \dots, N-1$  and  $T'(0) = \eta + \sum_{k=1}^{N-1} q'_k x'_k(0) + p'_N x'_N(0)$ . Set  $\alpha_1 = 1$ .
2. Compute  $u = \{u_k = u_k^0 + \Delta u_k\}_{k=0}^{N-1}$ ,  $x = \{x_{k+1} = H_k(x_k, u_k)\}_{k=0}^{N-1}$ , and  $f = f(x, u)$ .  $T(\alpha_1) = f$ .
3. If  $T(\alpha_1) \leq T(0) + c_1 T'(0)$  then stop with  $\alpha = \alpha_1$ .
4. Compute  $\alpha_{\min}$  by (5.48) and  $\alpha_2$  by (5.49).
5. Compute  $u = \{u_k = u_k^0 + \alpha_2 \Delta u_k\}_{k=0}^{N-1}$  and  $x = \{x_{k+1} = H_k(x_k, u_k)\}_{k=0}^{N-1}$ . Evaluate  $f = f(x, u)$  by (5.64a). Set  $T(\alpha_2) = f$ .
6. If  $T(\alpha_2) \leq T(0) + c_1 \alpha_2 T'(0)$  then stop with  $\alpha = \alpha_2$ .
7. Compute  $a$  and  $b$  by (5.51). If  $a = 0$  compute  $\alpha_{\min} = \frac{-T'(0)}{b}$ . Otherwise compute  $\alpha_{\min}$  by (5.52). Compute  $\alpha$  by (5.53).
8. Compute  $u = \{u_k = u_k^0 + \alpha_2 \Delta u_k\}_{k=0}^{N-1}$  and  $x = \{x_{k+1} = H_k(x_k, u_k)\}_{k=0}^{N-1}$ . Evaluate  $f = f(x, u)$  by (5.64a). Set  $T(\alpha) = f$ .
9. If  $T(\alpha) \leq T(0) + c_1 \alpha T'(0)$  then stop.
10. Set  $\alpha_1 = \alpha_2$ ,  $\alpha_2 = \alpha$ ,  $T(\alpha_1) = T(\alpha_2)$ , and  $T(\alpha_2) = T(\alpha)$ . Go to step (7).

The algorithm returns  $\alpha$ ,  $\{x_{k+1}, u_k\}_{k=0}^{N-1}$ , and  $f$ .

For unstable models the open-loop feasible path SQP method in terms of algorithm 10 and 11 may break down as the evaluations  $x_{k+1} = H_k(x_k, u_k)$  and  $f = \sum_{k=0}^{N-1} f_k(x_k, u_k) + f_N(x_N)$  may lead to numerical overflow. One immediately apparent method to avoid such situations is to monitor the objective function as well as the states and decrease the step length whenever divergence is detected. This requires that the initial state combined with the initial control trajectory  $\{u_k^0\}_{k=0}^{N-1}$  does not lead to divergence. Even if the system does not diverge, the trajectory generated by the feasible path open-loop SQP algorithm does not necessarily converge to a stationary solution (i.e. the optimal solution) when applied to an unstable system. The reason for non-convergence is that the algorithm ends up taking very small steps, i.e. the line search algorithms selects a very small step length. This is believed to be a manifestation of the Maratros effect (c.f. Maratros, 1978) and a consequence of the high sensitivity of the state trajectory,  $\{x_k\}_{k=0}^N$ , to small deviations in the controls,  $\{u_k\}_{k=0}^{N-1}$ .

### 5.4.2 Closed-Loop Feasible Path SQP

The closed-loop feasible path SQP algorithm ensures convergence even of unstable models by a state dependent modification of the input,  $u_k$ , computed in the forward run of the line search algorithm. This modification tackles both the divergence and the non-convergence problems encountered in the open-loop

feasible path SQP algorithm for optimal control. When the solution does not diverge but the model itself is unstable, the problems with the open-loop feasible path SQP algorithm is overcome by the closed-loop feasible path SQP as the nominal trajectory at each iteration is stabilized.

When  $\Delta u_k$  and  $\Delta x_k$  as well as  $K_k$  are computed by proposition 5.4.1, the input,  $u_k(\alpha)$ , may be computed by

$$\begin{aligned} u_k(\alpha) &= u_k^0 + \alpha \Delta u_k + K_k(x_k(\alpha) - [x_k^0 + \alpha \Delta x_k]) \\ &= u_k^0 + \alpha a_k + K_k(x_k(\alpha) - x_k^0) \end{aligned} \quad (5.136)$$

in which  $\alpha$  is the step length and  $a_k$  is computed according to proposition 5.4.1. The index has the range  $k = 0, 1, \dots, N-1$  and  $K_0 = 0$  by definition. With this expression for the input, a feasible closed-loop trajectory may be generated according to the recursion

$$u_0(\alpha) = u_0^0 + \alpha a_0 \quad (5.137a)$$

$$x_1(\alpha) = H_0(u_0(\alpha)) = H_0(x_0^0, u_0(\alpha)) \quad (5.137b)$$

$$u_k(\alpha) = u_k^0 + \alpha a_k + K_k(x_k(\alpha) - x_k^0) \quad (5.137c)$$

$$x_{k+1}(\alpha) = H_k(x_k(\alpha), u_k(\alpha)) \quad (5.137d)$$

in which  $k = 1, 2, \dots, N-1$ . As in the open-loop feasible path SQP method for optimal control, the merit function  $T(\alpha)$  in the closed-loop feasible path SQP algorithm for optimal control is given by (5.130). However, in the closed-loop feasible path SQP algorithm, the feasible trajectory used in evaluating the merit function is computed by (5.137). The step length,  $\alpha$ , is chosen such that it satisfies the Armijo condition. For the closed-loop feasible path SQP algorithm,  $T(0)$  and  $T'(0)$  used in verifying the Armijo condition are computed according to the following lemma.

**Lemma 5.4.3**

Let  $T : [0, 1] \mapsto \mathbb{R}$  be defined by (5.130). Let  $\{K_k\}_{k=1}^{N-1}$  and  $\{a_k\}_{k=0}^{N-1}$  be computed according to proposition 5.4.1. Let  $\{u_k(\alpha)\}_{k=0}^{N-1}$  and  $\{x_k(\alpha)\}_{k=1}^{N-1}$  be computed by (5.137). Let  $f^0$  be the value of (5.64a) at the nominal trajectory  $\{x_{k+1}^0, u_k^0\}_{k=0}^{N-1}$ . Let  $(r_0, \{q_k, r_k\}_{k=1}^{N-1}, p_N)$  be defined by (5.81) at the nominal trajectory  $\{x_{k+1}^0, u_k^0\}_{k=0}^{N-1}$ . Let  $(B_0, \{A_k, B_k\}_{k=1}^{N-1})$  be defined by (5.83a) and (5.84a)-(5.84b), respectively, at the nominal trajectory  $\{x_{k+1}^0, u_k^0\}_{k=0}^{N-1}$ .

Then

$$T(0) = \sum_{k=0}^{N-1} f_k(x_k^0, u_k^0) + f_N(x_N^0) = f^0 \quad (5.138)$$

and

$$T'(0) = r'_0 a_0 + \sum_{k=1}^{N-1} q'_k \frac{dx_k}{d\alpha}(0) + r'_k \frac{du_k}{d\alpha}(0) + p'_N \frac{dx_N}{d\alpha}(0) \quad (5.139)$$

in which  $\frac{dx_k}{d\alpha}(0)$  and  $\frac{du_k}{d\alpha}(0)$  are obtained by the recursion

$$\frac{dx_1}{d\alpha}(0) = B'_0 a_0 \quad (5.140a)$$

$$\frac{du_k}{d\alpha}(0) = a_k + K_k \frac{dx_k}{d\alpha}(0) \quad (5.140b)$$

$$\frac{dx_{k+1}}{d\alpha}(0) = A'_k \frac{dx_k}{d\alpha}(0) + B'_k \frac{du_k}{d\alpha}(0) \quad (5.140c)$$

for  $k = 1, 2, \dots, N-1$ .

*Proof.* (5.138) follows immediately from the fact that  $x_k(0) = x_k^0$  and  $u_k(0) = u_k^0$ . The expressions (5.139) and (5.140) are obtained by the chain rule of differentiation.  $\square$

By the above modifications of the feasible path SQP algorithm for optimal control, the closed-loop feasible path SQP algorithm may be formulated in terms of the statements in the algorithm below.

**Algorithm 12 (Feasible Path SQP for (5.64))**

Let  $x_0^0 = x_0$  and  $\{u_k^0\}_{k=0}^{N-1}$  be given.

1. Compute the initial state profile and objective function:  $\{x_{k+1}^0 = H_k(x_k^0, u_k^0)\}_{k=0}^{N-1}$  and  $f^0 = \sum_{k=0}^{N-1} f_k(x_k^0, u_k^0) + f_N(x_N^0)$ .
2. Evaluate the gradients along the nominal trajectory: Compute  $(r_0, \{q_k, r_k\}_{k=1}^{N-1}, p_N)$  by (5.81) and  $(B_0, \{A_k, B_k\}_{k=1}^{N-1})$  by (5.83a) and (5.84a)-(5.84b).
3. Compute an initial estimate of the Lagrange multipliers:  $\pi_{N-1}^0 = p_N$ , and  $\{\pi_{k-1}^0 = A_k \pi_k^0 - q_k\}_{k=N-1}^1$ .
4. Choose initial approximations for the Hessian block matrices:  $\{W_k\}_{k=0}^N$  are chosen as some symmetric positive definite matrices. Define  $R_0, \{Q_k, M_k, R_k\}_{k=1}^{N-1}$  and  $P_N$  according to (5.79).
5. Compute  $(\{\Delta x_{k+1}, \Delta u_k, \mu_k\}_{k=0}^{N-1})$ ,  $\{a_k\}_{k=0}^{M-1}$ , and  $\{K_k\}_{k=1}^{N-1}$  by proposition 5.4.1 using the data  $(R_0, r_0, B_0, \{Q_k, M_k, R_k, q_k, r_k, A_k, B_k\}_{k=1}^{N-1}, P_N, p_N)$ .
6. Check for convergence: Compute  $\Delta f = r'_0 \Delta u_0 + \sum_{k=1}^{N-1} q'_k \Delta x_k + r'_k \Delta u_k + p'_N \Delta x_N$ . If  $|\Delta f| \leq \epsilon$  then terminate with  $\{x_{k+1}^0, u_k^0, \mu_k\}_{k=0}^{N-1}$  as the solution. Return  $f^0$  as well.
7. Compute  $\alpha$  by the line search algorithm (algorithm 11). The result is  $\{x_{k+1}, u_k\}_{k=0}^{N-1}$  and  $f = f(x, u)$ .
8. Compute  $\{\pi_k = \pi_k^0 + \alpha(\mu_k - \pi_k^0)\}_{k=0}^{N-1}$ .
9. Compute the Lagrange derivatives using  $\{x_{k+1}^0, u_k^0, \pi_k\}_{k=0}^{N-1}$  (see step 7 of algorithm 9).
10. Evaluate the gradients along the trajectory  $\{x_{k+1}, u_k\}_{k=0}^{N-1}$ : Compute  $(r_0, \{q_k, r_k\}_{k=1}^{N-1}, p_N)$  by (5.81) and  $(B_0, \{A_k, B_k\}_{k=1}^{N-1})$  by (5.83a) and (5.84a)-(5.84b).
11. Compute the Lagrange derivatives using  $\{x_{k+1}, u_k, \pi_k\}_{k=0}^{N-1}$  (see step 9 of algorithm 9).

12. Compute  $\{s_k, y_k\}_{k=0}^N$  as in step 10 of algorithm 9.
13. Update the Hessian approximation as in step 11 of algorithm 9.
14. Update the old values:  $\{x_{k+1}^0, u_k^0, \pi_k^0\}_{k=0}^{N-1} \leftarrow \{x_{k+1}, u_k, \pi_k\}_{k=0}^{N-1}$  and  $f^0 \leftarrow f$ . Go to (5).

The major assumption in this particular formulation of the closed loop algorithm is that the initial state trajectory  $\{x_k^0\}_{k=1}^N$  associated with the initial state  $x_0 = x_0^0$  and the initial input trajectory  $\{u_k^0\}_{k=0}^{N-1}$  is non-divergent. If this restriction is undesirable, step (1) must be modified. Alternatively, one can adjust the algorithm such that the user of the algorithm must supply an initial feasible trajectory  $(x_0^0, \{x_{k+1}^0, u_k^0\}_{k=0}^{N-1})$ . By this modification, the first part of step (1) becomes redundant.

The fundamental change in the closed-loop feasible path algorithm compared to the open-loop feasible path algorithm is that  $\{K_k\}_{k=1}^{N-1}$  and  $\{a_k\}_{k=0}^{N-1}$  computed in step (5) of algorithm 12 by proposition 5.4.1 are used by the line search algorithm. The line search algorithm applies  $\{a_k\}_{k=0}^{N-1}$  and  $\{K_k\}_{k=1}^{N-1}$  in construction of the closed-loop trajectory along which the merit function is evaluated. With respect to the major principles, the line search algorithm presented below is identical with the previously presented line search algorithms.

**Algorithm 13 (Line Search for FSQP of (5.64))**

Let  $x^0 = \{x_k^0\}_{k=1}^N$  and  $u^0 = \{u_k^0\}_{k=0}^{N-1}$ . Let  $f^0 = f(x^0, u^0)$ ,  $\{x_{k+1}^0, u_k^0\}_{k=0}^{N-1}$ ,  $\{a_k\}_{k=0}^{N-1}$ ,  $\{K_k\}_{k=1}^{N-1}$ ,  $(\{r_k\}_{k=0}^{N-1}, \{q_k\}_{k=1}^{N-1}, p_N)$ , and  $(B_0, \{A_k, B_k\}_{k=1}^{N-1})$  be given. Let  $c_1 = 10^{-4}$ .

1. Set  $T(0) = f^0$ . Compute  $T'(0)$  by (5.139) using (5.140). Set  $\alpha_1 = 1$ .
2. Compute  $u(\alpha_1) = \{u_k(\alpha_1)\}_{k=0}^{N-1}$  and  $x(\alpha_1) = \{x_k(\alpha_1)\}_{k=1}^N$  by (5.137). Compute  $f = f(x(\alpha_1), u(\alpha_1))$  and set  $T(\alpha_1) = f$ .
3. If  $T(\alpha_1) \leq T(0) + c_1 T'(0)$  then stop with  $\alpha = \alpha_1$ .
4. Compute  $\alpha_{\min}$  by (5.48) and  $\alpha_2$  by (5.49).
5. Compute  $u(\alpha_2) = \{u_k(\alpha_2)\}_{k=0}^{N-1}$  and  $x(\alpha_2) = \{x_k(\alpha_2)\}_{k=1}^N$  by (5.137). Evaluate  $f = f(x(\alpha_2), u(\alpha_2))$  by (5.64a). Set  $T(\alpha_2) = f$ .
6. If  $T(\alpha_2) \leq T(0) + c_1 \alpha_2 T'(0)$  then stop with  $\alpha = \alpha_2$ .
7. Compute  $a$  and  $b$  by (5.51). If  $a = 0$  compute  $\alpha_{\min} = \frac{-T'(0)}{b}$ . Otherwise compute  $\alpha_{\min}$  by (5.52). Compute  $\alpha$  by (5.53).
8. Compute  $u(\alpha) = \{u_k(\alpha)\}_{k=0}^{N-1}$  and  $x(\alpha) = \{x_k(\alpha)\}_{k=1}^N$  by (5.137). Evaluate  $f = f(x(\alpha), u(\alpha))$  by (5.64a). Set  $T(\alpha) = f$ .
9. If  $T(\alpha) \leq T(0) + c_1 \alpha T'(0)$  then stop.
10. Set  $\alpha_1 = \alpha_2$ ,  $\alpha_2 = \alpha$ ,  $T(\alpha_1) = T(\alpha_2)$ , and  $T(\alpha_2) = T(\alpha)$ . Go to step (7).

The algorithm returns  $\alpha$ ,  $\{x_{k+1}(\alpha), u_k(\alpha)\}_{k=0}^{N-1}$ , and  $f$ .

## 5.5 Reduced-Space Quadratic Program

The quadratic program arising in both the infeasible and feasible path SQP algorithm described in the previous sections is solved using a factorization based on Riccati iteration. In this section an alternative for solution of the quadratic program (5.82) is presented. The method is based on elimination of the states and the constraints by formation of a smaller but dense quadratic program. In the first modification, the dense quadratic program is formed by elimination of the states in (5.82). In the second modification, the dense quadratic program is formed directly by implicit elimination of the states in (5.64) such that the resulting problem becomes an unconstrained optimization problem in the inputs only.

Algorithms solving a dense quadratic program rather than solving a structured quadratic programming using Riccati iteration may be advantageous when the system has much fewer inputs than states and a short prediction and control horizon compared to the state dimension.

### 5.5.1 Dense Quadratic Program

The structured quadratic program (5.82) may be transformed into a dense quadratic program by elimination of the states and the constraints representing the linearized dynamics. Consequently, rather than obtaining the solution by solving the structured quadratic program through Riccati iteration, it may be obtained by solving a dense quadratic program. This procedure may be incorporated in both the feasible path and infeasible path SQP algorithms for nonlinear optimal control.

The dense quadratic program is deduced by elimination of the constraints and states in the structured quadratic program (5.82). Elimination of the constraints

$$\Delta x_1 = B'_0 \Delta u_0 + b_0 \quad (5.141a)$$

$$\Delta x_{k+1} = A'_k \Delta x_k + B'_k \Delta u_k + b_k \quad k \in \mathcal{N} \quad (5.141b)$$

in the quadratic program (5.82) involves elimination of the states by expressing these as function of the inputs  $\{\Delta u_k\}_{k=0}^{N-1}$  and the exogeneous parameters  $\{b_k\}_{k=0}^{N-1}$ . These expressions are conveniently deduced and expressed using the discrete-time state transition matrix

$$\Phi_{k,j} = \Phi(k,j) = \begin{cases} A'_{k-1} A'_{k-2} \cdots A'_j & k > j \\ I & k = j \\ 0 & k < j \end{cases} \quad (5.142)$$

and the discrete-time impulse response matrix

$$\begin{aligned}\Gamma_{k,j} &= \Phi_{k,j+1} B'_j \\ &= \begin{cases} A'_{k-1} A'_{k-2} \cdots A'_{j+1} B'_j & k > j+1 \\ B'_j & k = j+1 \\ 0 & k < j+1 \end{cases} \end{aligned} \quad (5.143)$$

for  $k \in \{1, 2, \dots, N\}$  and  $j \in \{0, 1, \dots, N-1\}$ . Using the discrete-time state transition matrices and the discrete-time impulse response matrices, the dynamic equations (5.141) give the following expression for the states  $\{\Delta x_k\}_{k=1}^N$

$$\begin{aligned}\Delta x_k &= \sum_{j=0}^{k-1} \Gamma_{k,j} \Delta u_j + \sum_{j=0}^{k-1} \Phi_{k,j+1} b_j \\ &= \Gamma_k \Delta u + d_k\end{aligned} \quad (5.144)$$

The second part of this relation is expressed in terms of the decision vector  $\Delta u$  defined by  $\Delta u = [\Delta u'_0 \quad \Delta u'_1 \quad \dots \quad \Delta u'_{N-1}]'$  and the matrix  $\Gamma_k$  defined by

$$\begin{aligned}\Gamma_k &= [\Gamma_{k,0} \quad \Gamma_{k,1} \quad \dots \quad \Gamma_{k,k-1} \quad \Gamma_{k,k} \quad \dots \quad \Gamma_{k,N-1}] \\ &= [\Gamma_{k,0} \quad \Gamma_{k,1} \quad \dots \quad \Gamma_{k,k-1} \quad 0 \quad \dots \quad 0]\end{aligned} \quad (5.145)$$

The structure of  $\Gamma_k$  resembles to some extent the structure of the controllability matrix. The discrete-time impulse response matrix may be computed by the recursion

$$\begin{aligned}\Gamma_1 &= [\Gamma_{1,0} \quad \Gamma_{1,1} \quad \dots \quad \Gamma_{1,N-1}] \\ &= [B'_0 \quad 0 \quad \dots \quad 0]\end{aligned} \quad (5.146a)$$

$$\begin{aligned}\Gamma_{k+1} &= [\Gamma_{k+1,0} \quad \dots \quad \Gamma_{k+1,k-1} \quad \Gamma_{k+1,k} \quad 0 \quad \dots \quad 0] \\ &= [A'_k \Gamma_{k,0} \quad \dots \quad A'_k \Gamma_{k,k-1} \quad B'_k \quad 0 \quad \dots \quad 0]\end{aligned} \quad (5.146b)$$

The set of vectors,  $\{d_k\}_{k=1}^N$ , are defined by

$$d_k = \sum_{j=0}^{k-1} \Phi_{k,j+1} b_j \quad (5.147)$$

but more conveniently computed by the recursion

$$d_1 = b_0 \quad (5.148a)$$

$$d_{k+1} = A'_k d_k + b_k \quad k = 1, 2, \dots, N-1 \quad (5.148b)$$

With the quantities introduced in this section, the states and constraints can be eliminated from the quadratic program (5.82). The result is a quadratic program with  $\Delta u$  as the only decision variables. This result is stated in the following proposition.

**Proposition 5.5.1 (Dense Quadratic Program)**

Let  $\{\Gamma_k\}_{k=1}^N$  be defined by (5.145). Let  $\Phi_{k,j+1}$  and  $\Gamma_{k,j}$  be defined by (5.142) and (5.143) for  $k = \{1, 2, \dots, N\}$  and  $j = \{0, 1, \dots, N-1\}$ . Let  $\{d_k\}_{k=1}^N$  be computed by (5.148). Let  $\Delta u = [\Delta u'_0 \quad \Delta u'_1 \quad \dots \quad \Delta u'_{N-1}]'$ . Let  $I_k \in \mathbb{R}^{m \times Nm}$  denote a matrix with an identity matrix in block  $k$ , e.g.

$$I_k = [0 \quad \dots \quad 0 \quad I \quad 0 \quad \dots \quad 0] \quad (5.149)$$

Then the solution  $\{\Delta x_{k+1}^*, \Delta u_k^*\}_{k=0}^{N-1}$  of the quadratic program (5.82) may be obtained as the solution  $\{\Delta u_k^*\}_{k=0}^{N-1}$  of

$$\min_{\Delta u} \phi = \frac{1}{2} \Delta u' W \Delta u + r' \Delta u + \rho \quad (5.150)$$

and subsequent computation of  $\{\Delta x_k^*\}_{k=1}^N$ . The Hessian matrix,  $W$ , is given by

$$\begin{aligned} W = & \sum_{k=0}^{N-1} I'_k R_k I_k \\ & + \sum_{k=1}^{N-1} \Gamma'_k Q_k \Gamma_k + \Gamma'_k M_k I_k + I'_k M'_k \Gamma_k \\ & + \Gamma'_N P_N \Gamma_N \end{aligned} \quad (5.151)$$

and the gradient,  $r$ , is

$$\begin{aligned} r = & I'_0 r_0 \\ & + \sum_{k=1}^{N-1} \Gamma'_k (q_k + Q_k d_k) + I'_k (r_k + M'_k d_k) \\ & + \Gamma'_N (p_N + P_N d_N) \end{aligned} \quad (5.152)$$

The constant term,  $\rho$ , is

$$\rho = \sum_{k=1}^{N-1} \frac{1}{2} d'_k Q_k d_k + q'_k d_k + \frac{1}{2} d'_N P_N d_N + p'_N d_N \quad (5.153)$$

*Proof.* The quadratic program (5.150) with the data (5.151)-(5.153) is obtained from (5.82) by elimination of the states.  $\Delta x_k$  given by the expression (5.144) implies that the constraints (5.82b)-(5.82c) are always satisfied. Substitution of the expression (5.144) for the states in the objective function (5.80) and subsequent straightforward algebraic manipulations leads to the desired result.  $\square$

**Remark 5.5.2**

For unstable systems, the matrix  $W$  may be very ill-conditioned. Unstable modes corresponding to eigenvalues with absolute value larger than one may be amplified in the modified controllability matrices,  $\Gamma_k$ . As  $k$  becomes large the corresponding entries in the modified controllability matrices become very large. Consequently, by inspection of the expression for  $W$  it may have some directions with very large singular values.

The major computational efforts in solving the dense quadratic program (5.150) concerns computation of the reduced approximate Hessian matrix,  $W$ , by (5.151) and solution of the resulting quadratic program. The solution of (5.150) is stated in the following corollary.

**Corollary 5.5.3**

The solution  $\Delta u^*$  of the quadratic program (5.150) is obtained as the solution of the symmetric positive definite linear system

$$W\Delta u^* = -r \quad (5.154)$$

Subsequently,  $\{\Delta x_k^*\}_{k=1}^N$  can be obtained either from

$$\Delta x_k^* = \Gamma_k \Delta u^* + d_k \quad (5.155)$$

or from the recursion

$$\Delta x_1^* = B_0' \Delta u_0^* + b_0 \quad (5.156a)$$

$$\Delta x_{k+1}^* = A_k' \Delta x_k^* + B_k' \Delta u_k^* + b_k \quad (5.156b)$$

*Proof.*  $R_0, \begin{bmatrix} Q_k & M_k \\ M_k' & R_k \end{bmatrix}$  for  $k = 1, 2, \dots, N-1$ , and  $P_N$  are symmetric positive definite by construction. Hence,  $W$  computed by (5.151) is symmetric positive definite. Then the condition  $\nabla \phi = W\Delta u^* + r = 0$  is both necessary and sufficient for  $\Delta u^*$  being the unique global minimizer of (5.150). The expressions for computation of  $\{\Delta x_k^*\}_{k=1}^N$  follows directly from the preceding discussion in this section.  $\square$

The reduced approximate Hessian matrix,  $W$ , is a symmetric positive definite  $Nm \times Nm$  matrix. Hence, the solution of (5.154) may be obtained by Cholesky factorization,  $W = LL'$ , of  $W$ . This implies that the solution of the quadratic program (5.150), i.e. solution of (5.154), requires  $O((Nm)^3)$  floating point operations. The solution procedure based on the Riccati iteration has complexity  $O(Nm^3)$ , i.e. it is linear in the decision horizon. However, the number of floating point operations in the Riccati iteration based procedure is cubic in the state dimension. In comparison, the number of floating point operations in formation of the reduced dense Hessian matrix,  $W$ , and hence in obtaining the solution using the dense quadratic program is quadratic in the state dimension.

In a feasible path SQP algorithm for optimal control, the construction of the dense quadratic program simplifies as  $\{b_k = 0\}_{k=0}^{N-1}$ . The following corollary states these simplifications.

**Corollary 5.5.4**

Let the solution of the quadratic program (5.82) be part of the feasible path SQP algorithm for nonlinear optimal control.

Then  $\{b_k = 0\}_{k=0}^{N-1}$  and the computation of the reduced gradient,  $r$ , of (5.150) simplifies to

$$r = I_0' r_0 + \sum_{k=1}^{N-1} \Gamma_k' q_k + I_k' r_k + \Gamma_N' p_N \quad (5.157)$$



The constant,  $\rho$ , of (5.150) is  $\rho = 0$ .

The optimal state sequence,  $\{\Delta x_k^*\}_{k=1}^N$ , may be computed by

$$\Delta x_k^* = \Gamma_k \Delta u^* \quad (5.158)$$

or

$$\Delta x_1^* = B_0' \Delta u_0^* \quad (5.159a)$$

$$\Delta x_{k+1}^* = A_k' \Delta x_k^* + B_k' \Delta u_k^* \quad (5.159b)$$

*Proof.* It has already been established that  $\{b_k = 0\}_{k=0}^{N-1}$  in the feasible path SQP algorithm for nonlinear optimal control. This implies that  $\{d_k = 0\}_{k=1}^N$ . The simplified expression (5.157) is obtained from (5.152) by setting  $d_k = 0$ . Similarly, the expressions for the optimal state sequence  $\{\Delta x_k^*\}_{k=1}^N$  are obtained using  $d_k = 0$  and  $b_k = 0$ , respectively.  $\square$

In the infeasible path algorithm for optimal control as well as the feasible path algorithm for optimal control the Lagrange multipliers associated with (5.82) are needed. When the solution  $\{\Delta x_{k+1}^*, \Delta u_k^*\}_{k=0}^{N-1}$  is computed by solving the dense quadratic program and subsequent computation of the states, the Lagrange multipliers  $\{\mu_k^*\}_{k=0}^{N-1}$  may be computed according to the following corollary.

#### Corollary 5.5.5

Given the optimal solution  $\{\Delta x_{k+1}^*, \Delta u_k^*\}_{k=0}^{N-1}$  of (5.82), the corresponding optimal Lagrange multipliers  $\{\mu_k^*\}_{k=0}^{N-1}$  may be computed by the recursion

$$\mu_{N-1}^* = -P_N \Delta x_N^* - p_N \quad (5.160a)$$

$$\mu_{k-1}^* = A_k \mu_k^* - Q_k \Delta x_k^* - M_k \Delta u_k^* - q_k \quad (5.160b)$$

*Proof.* The result is just another formulation of corollary 5.3.4.  $\square$

### 5.5.2 Direct Formation of the Dense Quadratic Program

Instead of forming the dense quadratic program (5.150) from the structured quadratic program (5.82), the dense quadratic program may be formed directly in a feasible path SQP algorithm for (5.64). In this algorithm, the states and constraints are eliminated in the nonlinear program rather than in the structured quadratic program. By the constraints

$$x_1 = H_0(u_0) \quad (5.161a)$$

$$x_{k+1} = H_k(x_k, u_k) \quad k = 1, 2, \dots, N-1 \quad (5.161b)$$

the states,  $x = \{x_k = x_k(u)\}_{k=1}^N$ , may be regarded as functions of  $u = \{u_k\}_{k=0}^{N-1}$ . This relation may also be expressed as  $x = x(u)$ . Consequently, the objective

function (5.64a) may be regarded as a function,  $F(u)$ , of  $u$

$$\begin{aligned} F(u) &= f(x(u), u) \\ &= f_0(u_0) + \sum_{k=1}^{N-1} f_k(x_k(u), u_k) + f_N(x_N(u)) \end{aligned} \quad (5.162)$$

and the nonlinear discrete optimal control problem (5.64) is equivalent to the unconstrained optimization problem

$$\min_u F(u) \quad (5.163)$$

Quasi-Newton methods for solution of this unconstrained optimization problems compute the search direction using a quadratic approximation

$$\min_{\Delta u} \phi = \frac{1}{2} \Delta u' W \Delta u + \nabla F(u^0)' \Delta u + F(u^0) \quad (5.164)$$

of  $F(u)$  around the current point,  $u^0$ . The Hessian matrix,  $W$ , of  $F(u)$  is approximated by a variable metric secant update (c.f. Dennis and Schnabel, 1996). Usually, the BFGS update is employed. The gradient,  $\nabla F(u)$ , is needed in the formation of  $\phi$  and in the secant update of the BFGS or modified BFGS approximation,  $W$ , of the Hessian matrix. An expression for the gradient,  $\nabla F(u)$ , may be obtained by application of the chain rule or by noting the equivalence to construction of the dense quadratic program from the structured quadratic program in the feasible path algorithm. The gradient,  $\nabla F(u)$ , is given by the expression

$$\nabla F(u) = I_0' r_0 + \sum_{k=1}^{N-1} \Gamma_k' q_k + I_k' r_k + \Gamma_N' p_N \quad (5.165)$$

and is efficiently computed by the following algorithm.

**Algorithm 14 (Computation of  $\nabla F(u)$ )**

Let  $(x, u) = (x(u), u) = \{x_{k+1}(u), u_k\}_{k=0}^{N-1}$  be a given feasible trajectory.

1. Compute  $r_0 = \nabla_{u_0} f_0(u_0)$  and  $B_0 = \nabla_{u_0} H_0(u_0)$ . Set  $\nabla F = I_0' r_0$ . Set

$$\begin{aligned} \Gamma_1 &= [\Gamma_{1,0} \quad \Gamma_{1,1} \quad \dots \quad \Gamma_{1,N-1}] \\ &= [B_0' \quad 0 \quad \dots \quad 0] \end{aligned}$$

2. For  $k = 1, 2, \dots, N-1$  compute:  $q_k = \nabla_{x_k} f_k(x_k, u_k)$  and  $r_k = \nabla_{u_k} f_k(x_k, u_k)$ . Set  $\nabla F \leftarrow \nabla F + \Gamma_k' q_k + I_k' r_k$ . Compute  $A_k = \nabla_{x_k} H_k(x_k, u_k)$  and  $B_k = \nabla_{u_k} H_k(x_k, u_k)$ . Compute

$$\begin{aligned} \Gamma_{k+1} &= [\Gamma_{k+1,0} \quad \dots \quad \Gamma_{k+1,k-1} \quad \Gamma_{k+1,k} \quad 0 \quad \dots \quad 0] \\ &= [A_k' \Gamma_{k,0} \quad \dots \quad A_k' \Gamma_{k,k-1} \quad B_k' \quad 0 \quad \dots \quad 0] \end{aligned}$$

3. Compute  $p_N = \nabla_{x_N} f_N(x_N)$ . Set  $\nabla F \leftarrow \nabla F + \Gamma_N' p_N$ .

The major computational expense in algorithm 14 concerns construction of the discrete-time impulse response matrices,  $\Gamma_k$ . For dense systems, the constructions of these matrices from  $A_k$  and  $B_k$  require  $\frac{1}{2}(N^2 - N)n^2m$  floating point operations. Consequently, algorithm 14 has a complexity which is quadratic in the horizon,  $N$ , and the state dimension,  $n$ .

With this algorithm for computation of the gradient,  $\nabla F(u)$ , available, the nonlinear optimal control problem (5.64) may be solved by solving (5.163) using the following algorithm.

**Algorithm 15 (Reduced Feasible Path SQP for (5.64))**

Let  $u^0 = \{u_k^0\}_{k=0}^{N-1}$  and  $x_0^0 = x_0$  be given.

1. Compute the trajectory and objective value by  $\{x_{k+1}^0 = H_k(x_k^0, u_k^0)\}_{k=0}^{N-1}$  and  $F^0 = f_0(u_0^0) + \sum_{k=1}^{N-1} f_k(x_k^0, u_k^0) + f_N(x_N^0)$ .
2. Compute  $r^0 = \nabla F(u^0)$  using algorithm 14.
3. Choose an initial positive definite approximation,  $W$ , for the Hessian matrix  $\nabla^2 F(u^0)$ .
4. Compute  $\Delta u^*$  as the solution of the dense QP, i.e. by solving  $W\Delta u^* = -r^0$ .
5. Check for convergence. If  $|(r^0)' \Delta u^*| < \varepsilon$  then stop with  $\{x_{k+1}^0, u_k^0\}_{k=0}^{N-1}$  as the solution. Return  $F^0$  as well.
6. Compute the step length  $\alpha$  by a line search algorithm using Armijo's condition. The result is  $\alpha$ ,  $(x, u) = \{x_{k+1}, u_k\}_{k=0}^{N-1}$ , and  $F$ .
7. Compute  $r = \nabla F(u)$  by algorithm 14.
8. Update the approximate Hessian matrix,  $W$ , by Powell's BFGS modification. Compute  $s = u - u^0$ ,  $y = r - r^0$ , and

$$\theta = \begin{cases} 1 & s'y \geq 0.2s'Ws \\ \frac{0.8s'Ws}{s'Ws - s'y} & s'y < 0.2s'Ws \end{cases} \quad (5.166)$$

Compute  $z = \theta y + (1 - \theta)Ws$ . Compute  $W$  by the expression

$$W \leftarrow W - \frac{Wss'W}{s'Ws} + \frac{zz'}{s'z} \quad (5.167)$$

9. Update the old values:  $(x^0, u^0) = (x, u)$ ,  $F^0 = F$ , and  $r^0 = r$ . Go to (4).

Algorithm 15 constructs the approximate Hessian matrix,  $W$ , by the modified BFGS update. Hence, it is relatively straightforward to extend this algorithm to the constrained case. Furthermore, this ensures that the approximate Hessian,  $W$ , is always positive definite. As is evident, the construction of the approximate Hessian,  $W$ , by this algorithm is much simpler than in the algorithm constructing the dense quadratic program from the structured quadratic program. The solution of the quadratic program in each iteration is obtained by solving  $W\Delta u^* = -r^0$ . As  $W$  is symmetric and positive definite this solution procedure may be conducted by Cholesky factorization,  $W = LL'$ , of  $W$ . However, as  $W$  is obtained by a rank-one modification, the Cholesky factor  $L$  may

be obtained directly such that the factorization  $W = LL'$  becomes redundant (c.f. Dennis and Schnabel, 1996). Direct formation of the Cholesky factor of a BFGS modified matrix,  $W$ , is an  $O((Nm)^2)$  process while factorization of  $W$  is an  $O((Nm)^3)$  process. Consequently, a reduction in the computational operations corresponding to one order of magnitude can be expected by such a modification.

The line search algorithm employed in algorithm 15 uses the objective function and the Armijo condition in conjunction with backtracking to select the step length,  $\alpha$ .

**Algorithm 16 (Line search)**

Let  $x_0^0 = x_0$ ,  $(x^0, u^0) = \{x_{k+1}^0, u_k^0\}_{k=0}^{N-1}$ ,  $F^0 = F(u^0)$ ,  $r^0 = \nabla F(u^0)$ , and  $\Delta u^* = \{\Delta u_k^*\}_{k=0}^{N-1}$  be given.

1. Set  $\alpha_1 = 1$ . Set  $T(0) = F^0$  and  $T'(0) = (r^0)' \Delta u^*$ .
2. Compute  $u = \{u_k = u_k^0 + \Delta u_k^*\}_{k=0}^{N-1}$  and  $x = \{x_{k+1} = H_k(x_k, u_k)\}_{k=0}^{N-1}$ . Evaluate  $F = f(x, u)$  by (5.64a). Set  $T(\alpha_1) = F$ .
3. If  $T(\alpha_1) \leq T(0) + c_1 T'(0)$  then stop with  $\alpha = \alpha_1$ .
4. Compute  $\alpha_{\min}$  by (5.48) and  $\alpha_2$  by (5.49).
5. Compute  $u = \{u_k = u_k^0 + \alpha_2 \Delta u_k^*\}_{k=0}^{N-1}$  and  $x = \{x_{k+1} = H_k(x_k, u_k)\}_{k=0}^{N-1}$ . Evaluate  $F = f(x, u)$  by (5.64a). Set  $T(\alpha_2) = F$ .
6. If  $T(\alpha_2) \leq T(0) + c_1 \alpha_2 T'(0)$  then stop with  $\alpha = \alpha_2$ .
7. Compute  $a$  and  $b$  by (5.51). If  $a = 0$  compute  $\alpha_{\min} = \frac{-T'(0)}{b}$ . Otherwise compute  $\alpha_{\min}$  by (5.52). Compute  $\alpha$  by (5.53).
8. Compute  $u = \{u_k = u_k^0 + \alpha \Delta u_k^*\}_{k=0}^{N-1}$  and  $x = \{x_{k+1} = H_k(x_k, u_k)\}_{k=0}^{N-1}$ . Evaluate  $F = f(x, u)$  by (5.64a). Set  $T(\alpha) = F$ .
9. If  $T(\alpha) \leq T(0) + c_1 \alpha T'(0)$  then stop.
10. Set  $\alpha_1 = \alpha_2$ ,  $\alpha_2 = \alpha$ ,  $T(\alpha_1) = T(\alpha_2)$ , and  $T(\alpha_2) = T(\alpha)$ . Go to step (7).

Essentially, the above algorithm is an open-loop feasible path algorithm with all the deficiencies of such algorithms when applied to systems around an unstable equilibrium.

The quasi-Newton method for unconstrained optimization provided by Dennis and Schnabel (1996) employs the BFGS secant update rather than the modified BFGS secant update. To ensure positive definiteness of the sequence of Hessian approximations, they use the Armijo condition along with the condition

$$\nabla F(u^0 + \alpha \Delta u^*)' \Delta u^* \geq \beta \nabla F(u^0)' \Delta u^* \quad (5.168)$$

in which  $\beta \in (\alpha, 1)$ . Therefore, the line-search algorithm has to be modified as described by Dennis and Schnabel (1996). However, positive definiteness of the sequence of Hessian approximations is ensured in the algorithm presented by application of Powell's modified BFGS update. This feature makes the algorithm readily extensible to the case with inequality constraints.

## 5.6 Discrete Maximum Principle

This section presents an algorithm for solution of the optimal control problem (5.64) based on the discrete-time maximum principle. Even though the problem is a minimization problem the principle used in the solution will be referred to as the maximum principle and not the minimum principle. The presentation of the discrete-time maximum principle algorithm for solution of optimal control problems puts the SQP based algorithms in perspective in relation to classic methods for optimal control. Ravn (1999) provides a recent presentation of the discrete-time maximum principle and related algorithms.

The Lagrangian of (5.64) may be expressed as

$$\begin{aligned}
 \mathcal{L} &= f_0(u_0) + \sum_{k=1}^{N-1} f_k(x_k, u_k) + f_N(x_N) \\
 &\quad - \pi'_0(H_0(u_0) - x_1) \\
 &\quad - \sum_{k=1}^{N-1} \pi'_k(H_k(x_k, u_k) - x_{k+1}) \\
 &= f_0(u_0) - \pi'_0 H_0(u_0) \\
 &\quad + \sum_{k=1}^{N-1} f_k(x_k, u_k) - \pi'_k H_k(x_k, u_k) + \pi'_{k-1} x_k \\
 &\quad + f_N(x_N) + \pi'_{N-1} x_N \\
 &= \mathcal{H}_0(u_0, \pi_0) + \sum_{k=1}^{N-1} \mathcal{H}_k(x_k, u_k, \pi_k) + \pi'_{k-1} x_k \\
 &\quad + \mathcal{H}_N(x_N) + \pi'_{N-1} x_N
 \end{aligned} \tag{5.169}$$

in which the Hamiltonians,  $\mathcal{H}_k$  for  $k = 0, 1, \dots, N$ , are defined as

$$\mathcal{H}_0(u_0, \pi_0) = f_0(u_0) - \pi'_0 H_0(u_0) \tag{5.170a}$$

$$\mathcal{H}_k(x_k, u_k, \pi_k) = f_k(x_k, u_k) - \pi'_k H_k(x_k, u_k) \tag{5.170b}$$

$$\mathcal{H}_N(x_N) = f_N(x_N) \tag{5.170c}$$

Using the Hamiltonians, it is customary to refer to the Lagrange multipliers  $\pi_k$  as the co-state vectors. Furthermore, by comparison to (5.66) it is evident that  $\mathcal{L}_k$  and  $\mathcal{H}_k$  are related by

$$\mathcal{L}_0(u_0, \pi_0) = \mathcal{H}_0(u_0, \pi_0) \tag{5.171a}$$

$$\mathcal{L}_k(x_k, u_k, \pi_{k-1}, \pi_k) = \mathcal{H}_k(x_k, u_k, \pi_k) + \pi'_{k-1} x_k \tag{5.171b}$$

$$\mathcal{L}_N(x_N, \pi_{N-1}) = \mathcal{H}_N(x_N) + \pi'_{N-1} x_N \tag{5.171c}$$

and furthermore the block matrices (5.71) of the Hessian of the Lagrangian

may be expressed as

$$R_0 = \nabla_{u_0, u_0}^2 \mathcal{H}_0 \quad (5.172a)$$

$$\begin{bmatrix} Q_k & M_k \\ M'_k & R_k \end{bmatrix} = \begin{bmatrix} \nabla_{x_k, x_k}^2 \mathcal{H}_k & \nabla_{x_k, u_k}^2 \mathcal{H}_k \\ \nabla_{u_k, x_k}^2 \mathcal{H}_k & \nabla_{u_k, u_k}^2 \mathcal{H}_k \end{bmatrix} \quad (5.172b)$$

$$P_N = \nabla_{x_N, x_N}^2 \mathcal{H}_N \quad (5.172c)$$

This close correspondence between the Hamiltonians,  $\mathcal{H}_k$ , and the stage wise Lagrangians,  $\mathcal{L}_k$ , implies that the necessary optimality conditions (5.67) of (5.64) can be formulated in terms of the Hamiltonians,  $\mathcal{H}_k$ , rather than the stage wise Lagrangians,  $\mathcal{L}_k$ , or the Lagrangian,  $\mathcal{L}$ , itself. The first order necessary optimality conditions of (5.64) formulated in terms of the Hamiltonians,  $\mathcal{H}_k$ , are stated in the following proposition.

**Proposition 5.6.1**

Let  $\{x_{k+1}, u_k\}_{k=0}^{N-1}$  be a local minimizer of (5.64). Let the Hamiltonians,  $\mathcal{H}_k$  for  $k = 0, 1, \dots, N-1$ , be defined according to (5.170).

Then there exists unique Lagrange multipliers,  $\{\pi_k\}_{k=0}^{N-1}$ , such that

$$\nabla_{u_0} \mathcal{H}_0(u_0, \pi_0) = 0 \quad (5.173a)$$

$$x_1 = H_0(u_0) \quad (5.173b)$$

and for  $k = 1, 2, \dots, N-1$

$$\nabla_{u_k} \mathcal{H}_k(x_k, u_k, \pi_k) = 0 \quad (5.174a)$$

$$x_{k+1} = H_k(x_k, u_k) \quad (5.174b)$$

as well as

$$\pi_{N-1} = -\nabla_{x_N} \mathcal{H}_N(x_N) \quad (5.175a)$$

$$\pi_{k-1} = -\nabla_{x_k} \mathcal{H}_k(x_k, u_k, \pi_k) \quad (5.175b)$$

for  $k = 1, 2, \dots, N-1$ .

*Proof.* This is a direct specialization of proposition 5.2.2 to (5.64). Compare for instance also to (5.67). The linear independence constraint qualification is always satisfied for (5.64) which can be verified by inspection of (5.88).  $\square$

**Remark 5.6.2**

When the optimality conditions are expressed by the Hamiltonians, the Lagrange multipliers  $\pi_k$  are called the co-state vectors and (5.175) are called the adjoint equations.

Sequential quadratic programming algorithms obtain in principle the optimal solution to (5.64) by simultaneous solution of (5.173)-(5.175) using Newton's method. The maximum principle algorithm is based on sequential solution of (5.173)-(5.175). Given  $\{x_{k+1}^0, u_k^0\}_{k=0}^{N-1}$ , the co-state vectors  $\{\pi_k^0\}_{k=0}^{N-1}$  are

computed using the adjoint equations (5.175). Given  $\{x_k^0\}_{k=1}^N$  and  $\{\pi_k^0\}_{k=0}^{N-1}$ , a new optimal input candidate,  $\{u_k^*\}_{k=0}^{N-1}$ , is computed by solution of (5.173a) and (5.174a). This sequential solution procedure implies that the maximum achievable rate of convergence is linear. The maximum obtainable rate of convergence is quadratic using Newton's method and super-linear using a quasi-Newton methodology. However, compared to a Newton-based method, the main feature of the maximum-principle algorithm is its simplicity as is evident in the statement of the algorithm below.

**Algorithm 17 (Discrete Weak Maximum Principle)**

Let  $\{u_k^0\}_{k=0}^{N-1}$  and  $x_0^0 = x_0$  be given. Assume that the associated state trajectory  $\{x_k^0\}_{k=1}^N$  is non-divergent.

1. Compute  $\{x_{k+1}^0 = H_k(x_k^0, u_k^0)\}_{k=0}^{N-1}$ . Compute  $f^0 = \sum_{k=0}^{N-1} f_k(x_k^0, u_k^0) + f_N(x_N^0)$ .
2. Compute  $\pi_{N-1}^0 = -\nabla_{x_N} \mathcal{H}_N(x_N^0)$  and  $\{\pi_{k-1}^0 = -\nabla_{x_k} \mathcal{H}_k(x_k^0, u_k^0, \pi_k^0)\}_{k=1}^{N-1}$ .
3. Compute  $\{u_k^*\}_{k=0}^{N-1}$  by solution of  $\nabla_{u_0} \mathcal{H}_0(u_0^*, \pi_0^0) = 0$  and  $\nabla_{u_k} \mathcal{H}_k(x_k^0, u_k^*, \pi_k^0) = 0$ . Compute  $\{\Delta u_k = u_k^* - u_k^0\}_{k=0}^{N-1}$ .
4. Check for convergence. If  $\|\Delta u_k\|_2 \leq \varepsilon$  for  $k = 0, 1, \dots, N-1$  then terminate with  $\{x_{k+1}^0, u_k^0, \pi_k^0\}_{k=0}^{N-1}$  as the solution. Return  $f^0$  as well.
5. Compute the step length  $\alpha$  by line search using the Armijo condition and

$$\{u_k(\alpha) = u_k^0 + \alpha \Delta u_k\}_{k=0}^{N-1} \quad (5.176a)$$

$$\{x_{k+1}(\alpha) = H_k(x_k(\alpha), u_k(\alpha))\}_{k=0}^{N-1} \quad (5.176b)$$

in computation of the merit function

$$T(\alpha) = \sum_{k=0}^{N-1} f_k(x_k(\alpha), u_k(\alpha)) + f_N(x_N(\alpha)) \quad (5.177)$$

6. Set  $\{u_k^0 = u_k(\alpha)\}_{k=0}^{N-1}$ ,  $\{x_{k+1}(\alpha) = x_{k+1}^0\}$  and  $f^0 = T(\alpha)$ . Go to (2).

**Remark 5.6.3**

When  $R_k = \nabla_{u_k, u_k}^2 \mathcal{H}_k(x_k^0, u_k, \pi_k^0)$  is positive definite in a neighborhood of  $u_k^*$ , the solution of

$$\nabla_{u_k} \mathcal{H}_k(x_k^0, u_k^*, \pi_k^0) = 0 \quad (5.178)$$

is equivalent to the minimization

$$\min_{u_k \in \mathbb{R}^m} \mathcal{H}_k(x_k^0, u_k, \pi_k^0) \quad (5.179)$$

This is the reason why the algorithm is called the maximum (minimum) principle algorithm. At each stage the corresponding Hamiltonian,  $\mathcal{H}_k$ , is minimized with respect to  $u_k$ .

Let  $R_k = \nabla_{u_k, u_k}^2 \mathcal{H}_k(x_k^0, u_k, \pi_k^0)$  and assume that this matrix is positive definite. Then the solution of (5.179) may be obtained iteratively by solution of the following equation to obtain the direction  $\Delta u_k^*$  for a new iterate

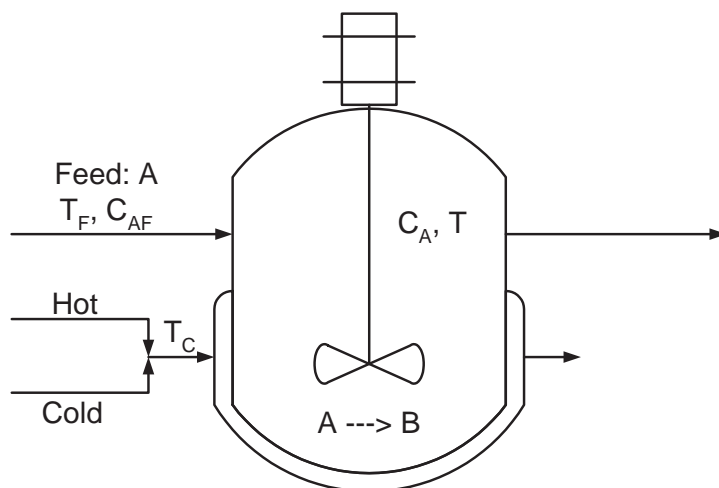
$$R_k \Delta u_k^* = -\nabla_{u_k} \mathcal{H}_k(x_k^0, u_k, \pi_k^0) \quad (5.180)$$

The matrices  $R_k$  may also be constructed using the BFGS-approximation such that (5.179) is solved by a quasi-Newton method.

The stated maximum principle algorithm is a feasible path open-loop method. This implies that it is not well suited for problems with unstable dynamics. Compared to the SQP based methods for solution of optimal control problems, the maximum principle algorithm requires less memory as it does not have to store the entire KKT-matrix. It is also fairly simple to program, but can only achieve linear rate of convergence. Another disadvantage is that it is not easily extensible to problems with state constraints.

## 5.7 Process Example

The potential problems with dynamic optimization of models with unstable dynamics is illustrated using a simple continuous stirred tank reactor with an exothermic irreversible reaction,  $A \rightarrow B$  (c.f. Henson and Seborg, 1997; Seborg *et al.*, 1989). The system is illustrated in figure 5.1. Reactant  $A$  at temperature  $T_f$  is fed to a stirred vessel with a cooling/heating jacket. The objective is to control the temperature,  $T$ , of the vessel by manipulating the temperature,  $T_c$ , of the fluid in the jacket. The manipulated jacket fluid temperature,  $T_c$ , is in practice a setpoint to a controller controlling the jacket temperature by manipulation of the flow rate of cold and hot jacket fluid. The jacket fluid temperature dynamics is assumed instantaneous relative to the material and energy dynamics of the stirred vessel.



**Figure 5.1.** An exothermic CSTR with a cooling jacket.



The material and energy balance of the constant volume CSTR is

$$\dot{C}_A = \frac{q}{V} (C_{A,f} - C_A) - r(C_A, T) \quad (5.181a)$$

$$\dot{T} = \frac{q}{V} (T_f - T) + \frac{(-\Delta H)}{\rho C_p} r(C_A, T) + \frac{UA}{V\rho C_p} (T_c - T) \quad (5.181b)$$

in which the reaction rate,  $r(C_A, T)$ , is first order in  $A$  and governed by an Arrhenius temperature dependence

$$r(C_A, T) = k(T)C_A, \quad k(T) = k_0 \exp\left(-\frac{E}{RT}\right) \quad (5.182)$$

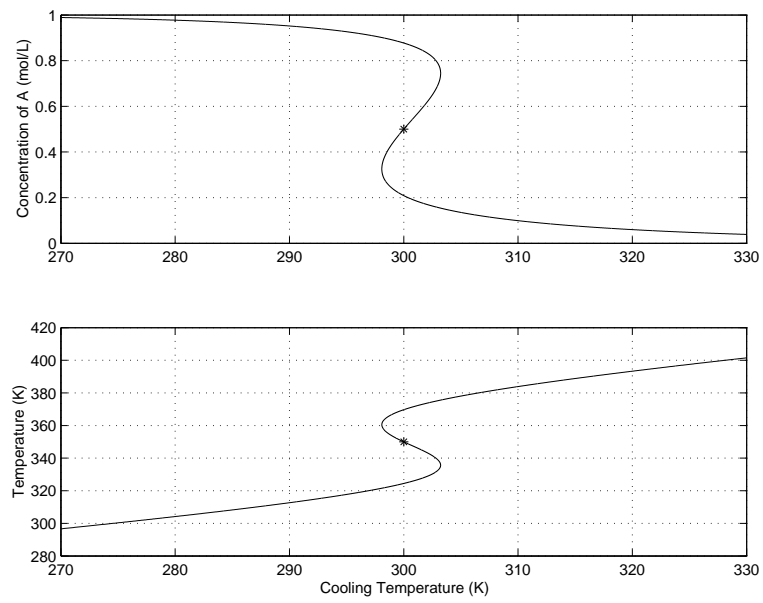
$C_A$  is the concentration of  $A$ ,  $q$  is the flow rate to and from the reactor,  $V$  is the volume of liquid in the reactor,  $C_{A,f}$  is the feed concentration of reactant  $A$ ,  $T$  is the temperature of the liquid in the reactor,  $T_f$  is the temperature of the feed,  $\Delta H$  is the reaction enthalpy,  $\rho$  is the density of the liquid,  $C_p$  is the specific heat capacity,  $UA$  is the heat transmission number between the jacket and the vessel, and  $T_c$  is the temperature of the jacket fluid. The parameters and the nominal operating point of the CSTR in figure 5.1 are provided in table 5.2.

The steady states of the system as function of the jacket fluid temperature,  $T_c$ , is plotted in figure 5.2. The asterisk indicates the nominal operating point. At the nominal jacket fluid temperature, multiple equilibria exist and the nominal equilibrium is unstable. Among other things, the instability implies that the steady state reactor temperature,  $T$ , and concentration,  $C_A$ , are highly sensitive to the actual value of the fluid temperature in the jacket.

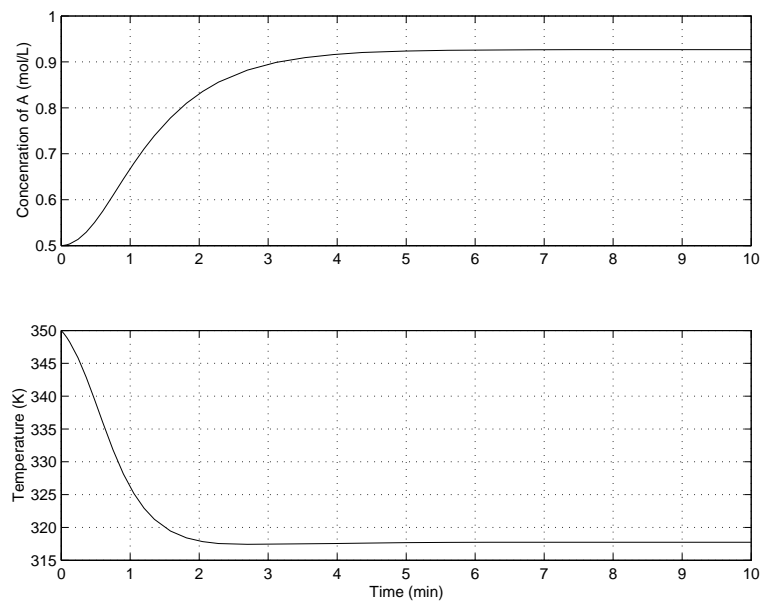
A decrease in the jacket fluid temperature from  $T_c = 300$  K to  $T_c = 295$  K implies that the reaction turns off and moves to the lower branch of the temperature curve. This transient behavior is shown in figure 5.3. The opposite situation in which the jacket fluid temperature is increased from  $T_c = 300$  K to  $T_c = 305$  K is illustrated in figure 5.4. The temperature increase ignites the reaction and the system goes into a limit cycle. As a result, excessive temperature and concentration variations are observed. The center point of the limit

**Table 5.2.** Parameters and nominal operating conditions for the CSTR.

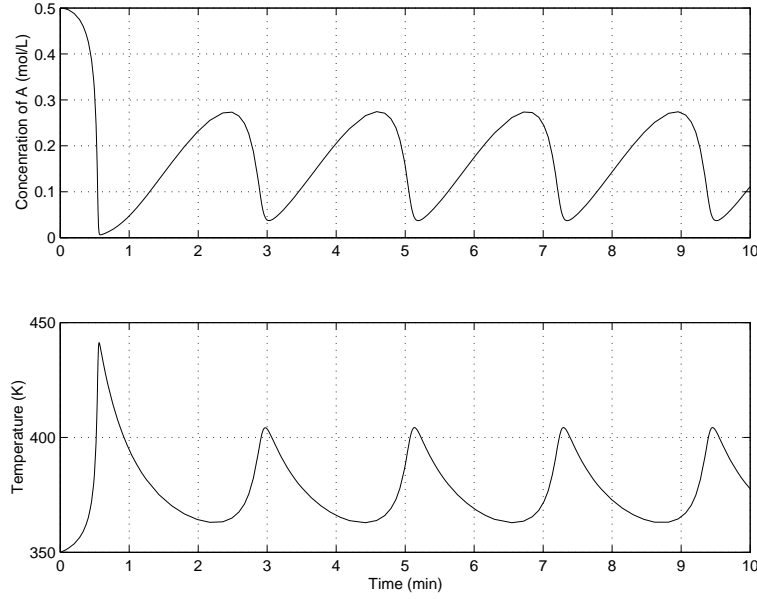
| Variable      | Value                        | Variable      | Value                                     |
|---------------|------------------------------|---------------|---|
| $q$           | 100 L/min                    | $\frac{E}{R}$ | 8750 K                                    |
| $C_{A,f}$     | 1.0 mol/L                    | $k_0$         | $7.2 \cdot 10^{10} \text{ min}^{-1}$      |
| $T_f$         | 350 K                        | $UA$          | $5 \cdot 10^4 \text{ J/min}\cdot\text{K}$ |
| $V$           | 100 L                        | $T_c$         | 300 K                                     |
| $\rho$        | 1000 g/L                     | $C_A$         | 0.5 mol/L                                 |
| $C_p$         | 0.239 J/g·K                  | $T$           | 350 K                                     |
| $(-\Delta H)$ | $5 \cdot 10^4 \text{ J/mol}$ |               |   |



**Figure 5.2.** Steady states of the exothermic CSTR.



**Figure 5.3.** A small decrease in the jacket fluid temperature ( $T_c = 295$  K) turns the reaction off.



**Figure 5.4.** A small increase in the jacket fluid temperature ( $T_c = 305$  K) ignites the reaction and drives the system into a limit cycle.

cycle corresponds to a point on the upper branch of the temperature steady state curve.

As is evident the system is highly nonlinear and unstable around the nominal operating point. The objective function used for driving the system to its feasible steady state  $(x_s, u_s)$  is

$$\begin{aligned} \psi = \frac{1}{2} \int_{t_0}^{t_f} & (x(t) - x_s)' Q (x(t) - x_s) \\ & + (u(t) - u_s)' R (u(t) - u_s) dt \\ & + \frac{1}{2} (x(t_f) - x_s)' P (x(t_f) - x_s) \end{aligned} \quad (5.183)$$

in which  $x = (C_A \quad T)'$  and  $u = T_c$ . The steady state used is the nominal operating point

$$x_s = \begin{bmatrix} 0.5 \\ 350 \end{bmatrix} \quad u_s = 300$$

and the selected weight matrices are<sup>1</sup>

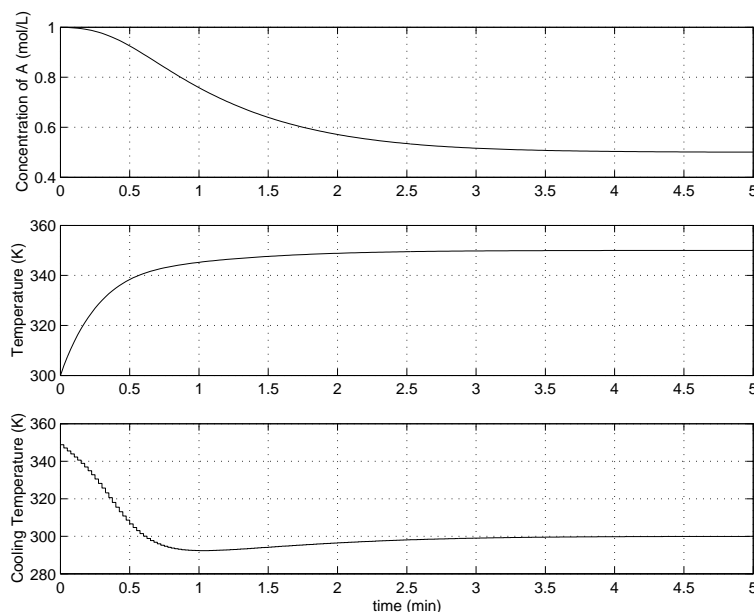
$$Q = \begin{bmatrix} 0 & 0 \\ 0 & 4 \end{bmatrix} \quad R = 8 \quad P = \begin{bmatrix} 99165 & 2104 \\ 2104 & 73 \end{bmatrix}$$

<sup>1</sup>Arbitrarily,  $P$  is selected to be the same as the one used by Tenny and Rawlings (2002). It should be computed by solution of an appropriate Riccati equation.

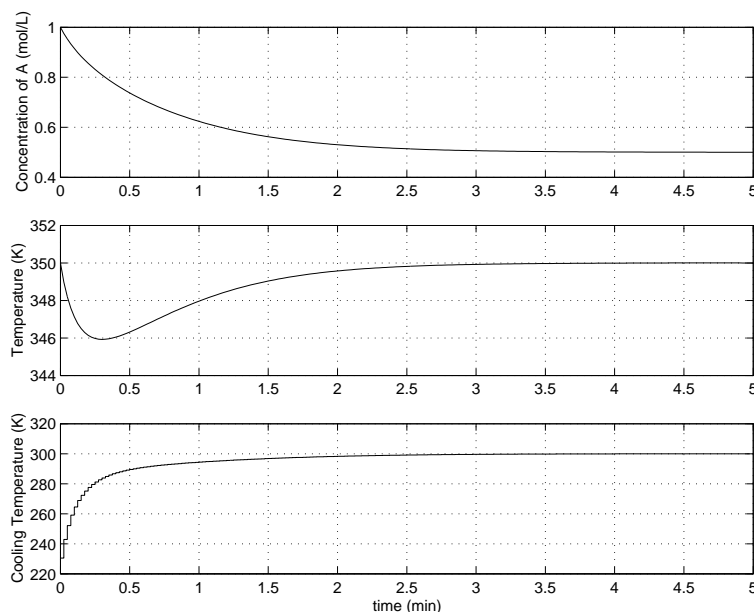
The optimal control problem used in controlling the temperature consists of minimizing (5.183) subject to the physics of the system (5.181). Figure 5.5 shows an optimal start-up trajectory computed by solution of the optimal control problem for the CSTR. The initial state is a full reactor with cold feed, i.e.  $C_A = C_{A,f} = 1.0$  mol/L and  $T = 300$  K. Initially, the computed jacket fluid temperature is warmer than the reactor content to ignite the reaction. Subsequently, the jacket fluid temperature is decreased and colder than the reactor content to avoid ignition and remove the heat produced by reaction. Physically and intuitively, the computed optimal trajectory makes good intuitive sense.

Figure 5.6 shows a stabilizing trajectory for a critical state, (1.0 mol/L, 350 K). Left uncontrolled with the nominal jacket fluid temperature this state of the reactor would ignite the reaction and lead to a run-away situation. As expected from physical intuition, the optimal stabilizing trajectory computed by the optimal control problem avoids run-away by severe cooling. This removes the generated heat and smoothly brings the system back to its desired unstable equilibrium point.

From figure 5.5 and figure 5.6 it is apparent that the nonlinear optimal controller is able to generate trajectories that are very appealing from a physical point of view even though the system has severe nonlinearities in the operating windows considered.



**Figure 5.5.** Optimal startup of the exothermic CSTR.



**Figure 5.6.** Stabilization of a critical state of the exothermic CSTR.

Both the infeasible path SQP method and the closed-loop feasible path SQP method are able to produce the trajectories shown in figure 5.5 and figure 5.6. However, the open-loop feasible path SQP method fails to converge. In some cases, it fails due to a crude fixed step size in the explicit Euler discretization such that the trajectories diverge. In other cases, when the step size used by the Euler integration is sufficiently small such that divergence due to the discretization is avoided, the open-loop feasible path method still fails to converge even though the generated trajectories to some extent approach the optimal trajectories. This phenomenon is believed to be a manifestation of the Maratros effect associated with Powell's  $l_1$ -merit function (c.f. Maratros, 1978).

## 5.8 Extensions and Refinements

To focus on the *principles* for solution of the nonlinear optimal control problem, this paper has been concerned with algorithms for the *unconstrained* nonlinear optimal control problem using Euler integration. This section describes extensions of the algorithms presented to constrained nonlinear optimal control problems, discuss the choice of merit function and efficient approximation of the Hessian matrix when the nonlinear optimal control problem is of the least squares type. Furthermore, efficient integration methods along with sensitivity

computation, their extension to index-1 differential algebraic equations, and their parallelization are discussed. All these extensions are facilitated by a thorough understanding of the unconstrained nonlinear optimal control problem.

### 5.8.1 Inequality Constraints

It is important to be able to extend the algorithms for solution of unconstrained optimal control problems to constrained optimal control problems. Almost all practical control problems are constrained as there is at least limits on the actuators. Often the rate of movement is also constrained as well as it is commonplace to use general state dependent functionalities to remain in some operation regime.

The issues involved in extending the algorithms for unconstrained nonlinear optimal control problems to constrained nonlinear optimal control problems are best explained using the generic nonlinear optimization problem (5.1) as a template. The extension of the generic equality constrained optimization problem (5.1) with inequality constraints is

$$\min_{x \in \mathbb{R}^n} f(x) \quad (5.184a)$$

$$s.t. \quad g_i(x) = 0 \quad i \in \mathcal{E} \quad (5.184b)$$

$$h_i(x) \geq 0 \quad i \in \mathcal{I} \quad (5.184c)$$

The Lagrangian function associated with (5.184) is

$$\mathcal{L}(x, \pi, \lambda) = f(x) - \pi' g(x) - \lambda' h(x) \quad (5.185)$$

and the partial derivative with respect to the primal variables is

$$\nabla_x \mathcal{L}(x, \pi, \lambda) = \nabla f(x) - \nabla g(x) \pi - \nabla h(x) \lambda \quad (5.186)$$

By this expression, it is evident how to evaluate the partial derivative of the Lagrangian with respect to the primal variables. This expression is applied in updating the variable metric approximation,  $W$ , of the Hessian matrix. The necessary optimality conditions may be stated using the complementarity conditions

$$\nabla_x \mathcal{L} = \nabla f(x) - \nabla g(x) \pi - \nabla h(x) \lambda = 0 \quad (5.187a)$$

$$\nabla_\pi \mathcal{L} = -g(x) = 0 \quad (5.187b)$$

$$\nabla_\lambda \mathcal{L} = -h(x) \leq 0 \quad (5.187c)$$

$$\lambda \geq 0 \quad (5.187d)$$

$$\lambda_i h_i(x) = 0 \quad i \in \mathcal{I} \quad (5.187e)$$

or alternatively they may be stated using the active set formulation

$$\nabla_x \mathcal{L} = \nabla f(x) - \nabla g(x)\pi - \nabla h(x)\lambda = 0 \quad (5.188a)$$

$$\nabla_\pi \mathcal{L} = -g(x) = 0 \quad (5.188b)$$

$$\nabla_\lambda \mathcal{L} = -h(x) \leq 0 \quad (5.188c)$$

$$\lambda_i \geq 0 \quad i \in \mathcal{A}(x) = \{i \in \mathcal{I} : h_i(x) = 0\} \quad (5.188d)$$

$$\lambda_i = 0 \quad i \notin \mathcal{A}(x) = \{i \in \mathcal{I} : h_i(x) = 0\} \quad (5.188e)$$

The presence of inequality constraints adds significant complexity to the procedure of obtaining the solution by solving the optimality conditions. However, as in the case without inequalities, the solution of the optimality conditions may be obtained by a Newton type procedure in which the search direction is obtained by quadratic programming. The quadratic program obtained in solving (5.184) is

$$\min_{\Delta x \in \mathbb{R}^n} \quad \phi = \frac{1}{2} \Delta x' W \Delta x + \nabla f(x^0)' \Delta x \quad (5.189a)$$

$$s.t. \quad g_i(x^0) + \nabla g_i(x^0)' \Delta x = 0 \quad i \in \mathcal{E} \quad (5.189b)$$

$$h_i(x^0) + \nabla h_i(x^0)' \Delta x \geq 0 \quad i \in \mathcal{I} \quad (5.189c)$$

The various methods for solution of this problem and the quadratic program corresponding to the optimal control problem are discussed by Jørgensen *et al.* (2004). The solution of the inequality constrained quadratic is somewhat more demanding from a computational point of view compared to the solution of the quadratic program with equalities only. The reason is that the solution must be obtained iteratively in which a sequence of problems corresponding to equality constrained quadratic programs are solved.

Besides the inequalities in the quadratic program and the resulting modifications in the procedure for obtaining the search direction, the other major consequence of adding inequalities to the nonlinear optimization problem concerns the merit function. Powell's exact  $l_1$ -penalty function associated with (5.184) is

$$\begin{aligned} P(x, \sigma, \tau) = f(x) &+ \sum_{i \in \mathcal{E}} \sigma_i |g_i(x)| \\ &+ \sum_{i \in \mathcal{I}} \tau_i |\min\{0, h_i(x)\}| \end{aligned} \quad (5.190)$$

The terms

$$|\min\{0, h_i(x)\}| = \begin{cases} 0 & h_i(x) \geq 0 \\ -h_i(x) & h_i(x) < 0 \end{cases} \quad (5.191)$$

are due to the inequalities and penalize any violation of the inequalities. The corresponding terms in the merit function may be denoted

$$\eta_i(\alpha) = |\min\{0, h_i(x^0 + \alpha \Delta x)\}| \quad i \in \mathcal{I} \quad (5.192)$$

Hence, the corresponding quantities required by the Armijo condition are

$$\eta_i(0) = \begin{cases} 0 & h_i(x^0) \geq 0 \\ -h_i(x^0) & h_i(x^0) < 0 \end{cases} \quad (5.193a)$$

$$\frac{d\eta_i}{d\alpha}(0) = \begin{cases} 0 & h_i(x^0) \geq 0 \\ -\nabla h_i(x^0)' \Delta x & h_i(x^0) < 0 \end{cases} \quad (5.193b)$$

in which the fact that  $\Delta x$  is obtained as the solution of (5.189) has been utilized. The penalty parameters,  $\sigma_i$  and  $\tau_i$ , are updated in a way analogously to the case without inequality constraints

$$\sigma_i \leftarrow \max\{|\mu_i|, \frac{1}{2}(\sigma_i + |\mu_i|)\} \quad (5.194a)$$

$$\tau_i \leftarrow \max\{\kappa_i, \frac{1}{2}(\tau_i + \kappa_i)\} \quad (5.194b)$$

$\mu_i$  are the Lagrange multipliers associated with the equality constraints in the quadratic program (5.189) and  $\kappa_i$  are the Lagrange multipliers associated with the inequality constraints in the quadratic program (5.189).

For the constrained general optimization problems and inequality constrained optimal control problems, the possibility of infeasibilities due to the problem statement or due to an inconsistent linearization must be considered by the algorithm. Biegler and Cuthrell (1985) and Gill *et al.* (1997) discuss these considerations from an algorithmic point of view, while Scokaert and Rawlings (1999) consider infeasibility issues from a model predictive control application point of view.

The nonlinear optimal control problem with general constraints may be stated as

$$\min_{x(t), u(t)} \psi = \int_{t_0}^{t_f} l(x(t), u(t)) dt + L(x(t_f), t_f) \quad (5.195a)$$

$$s.t. \quad \dot{x}(t) = h(x(t), u(t), t) \quad (5.195b)$$

$$x(t_0) = x_0 \quad (5.195c)$$

$$g(x(t), u(t), t) \geq 0 \quad (5.195d)$$

The extension compared to the unconstrained optimal control problem (5.54) is the general constraint,  $g(x(t), u(t), t) \geq 0$ , which must be enforced at all times,  $t$ , in the interval  $[t_0, t_f]$ . For most practical applications, the constraints are much more structured than indicated by the general constraints. For instance, typically a subset of the constraints are actuator limits  $u_{\min} \leq u(t) \leq u_{\max}$ . The constrained optimal control problem is discretized in the same way as the



unconstrained optimal control problem

$$\min_{x,u} \quad f(x,u) = \sum_{k=0}^{N-1} f_k(x_k, u_k) + f_N(x_N) \quad (5.196a)$$

$$s.t. \quad x_{k+1} = H_k(x_k, u_k) \quad k = 0, 1, \dots, N-1 \quad (5.196b)$$

$$G_k(x_k, u_k) \geq 0 \quad k = 0, 1, \dots, N-1 \quad (5.196c)$$

in which  $x_0$  is a parameter and not a decision variable. This discretization constitutes an approximation of the constrained continuous-time optimal control problem (5.195). One approximation is that the general function  $u(t)$  is restricted to belong to the space of piecewise constant functions. In the continuous time problem, the inequality constraint must be enforced at all time while in the discrete time case it is only enforced at discrete times. Depending on the discretization procedure applied to the differential equations, the inequalities may be violated at times intermediate to the discrete times at which the inequalities are enforced.

However, (5.196) is accepted as a reasonable approximation of (5.195) reflecting the requirements for a practical solution. Consequently, the algorithms for the unconstrained nonlinear optimal control problem (5.54) may be extended to the constrained nonlinear optimal control problem (5.196) using the algorithmic principles outlined for extending the general equality constrained optimization problem with inequality constraints. In approximating the Hessian matrix it is important to note that the partial separability of the Lagrangian function is retained as the inequality constraints each involve variables from one time period only. Consequently, the structure of the Hessian matrix observed for the unconstrained optimal control problem is retained for the constrained optimal control problem. This implies that the resulting quadratic program that must be solved has a structure that can be utilized in its efficient solution (c.f. Jørgensen *et al.*, 2004).

### 5.8.2 Merit Function

The merit function is another component of the infeasible path SQP algorithm that can be modified to enhance the performance of the algorithm. As has already been mentioned several times, Powell's exact  $l_1$  penalty function may suffer from the Marattos effect. To overcome this deficiency, Schittkowski (1981a) proposed an augmented Lagrangian penalty function

$$P(x, \pi) = \underbrace{f(x) - \pi'g(x)}_{=\mathcal{L}(x, \pi)} + \frac{1}{2} \sum_{j=1}^m \sigma g_j(x)^2 \quad (5.197)$$

to determine the step length in an SQP algorithm for the general equality constrained nonlinear program (5.1). In this expression,  $\sigma$  is a positive penalty

parameter that serves to penalize constraint violations and stabilizes the unconstrained line search minimization in the sense that it prevents construction of unbounded and therefore unsolvable subproblems. Schittkowski (1981a) presented his penalty function

$$\begin{aligned}
 P(x, \pi, \lambda) = & f(x) - \pi'g(x) + \frac{1}{2} \sum_{j \in \mathcal{E}} \sigma g_j(x)^2 \\
 & - \sum_{j \in \mathcal{I}} \begin{cases} \lambda_j h_j(x) - \frac{1}{2} \sigma h_j(x)^2 & h_j(x) \leq \frac{\lambda_j}{\sigma} \\ \frac{1}{2} \frac{\lambda_j^2}{\sigma} & h_j(x) > \frac{\lambda_j}{\sigma} \end{cases} \quad (5.198)
 \end{aligned}$$

to the constrained general optimization problem (5.184) (c.f. Bertsekas, 1995b, section 4.2). As explained by Bertsekas (1995b), the last part of this penalty function is obtained through introduction of slack variables and conversion to an equality constrained optimization problem. The advantage of this penalty function is that it unlike Powell's exact  $l_1$ -penalty is differentiable. This implies that algorithms computing the step length using this penalty function do not suffer from the Maratros effect. Schittkowski (1981a) also provides an algorithm for updating the penalty parameter  $\sigma$  such that convergence is guaranteed. Gill *et al.* (1992, 1997) used a similar penalty function based on slack variables but applied different penalty parameters to each constraint. Eldersveld (1991) provides the detailed updating procedure of the penalty parameters employed by Gill *et al.* (1997). The augmented Lagrangian merit function for selection of the step length was also incorporated in the algorithm developed by Biegler and Cuthrell (1985). A slight variant of the augmented Lagrangian penalty function has been proposed and applied by Fletcher (c.f. Nocedal and Wright, 1999; Bertsekas, 1995b) to avoid the Maratros effect. The difference in Fletcher's method compared to the method applied by Schittkowski (1981a) and Gill *et al.* (1997) is that Fletcher computes the Lagrange multipliers as the least squares estimate. This is an extra computational burden compared to the method presented by Schittkowski (1981a).

The Maratros effect can also be avoided by application of the watchdog technique (c.f. Chamberlain *et al.*, 1982; Hoza and Stadtherr, 1993; Nocedal and Wright, 1999) in conjunction with Powell's exact  $l_1$  penalty function. The watchdog technique is a non-monotone strategy allowing iterates that occasionally increase the merit function if a sufficient decrease in the merit function is obtained within a certain number of iterates.

### 5.8.3 Hessian Approximations

The least squares objective function is often used in control problems when some variables have to be tracked. From a numerical point of view, the least squares objective function has special features that can be used in the efficient construction of the Hessian matrix approximation (c.f. Jørgensen *et al.*, 2002).

The continuous-time least squares objective function of the nonlinear optimal control problem (5.54) is

$$\begin{aligned} \psi = & \frac{1}{2} \int_{t_0}^{t_f} \|\varphi(x(t), u(t), t)\|_{\Lambda(t)}^2 dt \\ & + \frac{1}{2} \|\varphi(x(t_f), t_f)\|_{\Lambda(t_f)}^2 \end{aligned} \quad (5.199)$$

in which  $\varphi(x(t), u(t), t)$  and  $\varphi(x(t_f), t_f)$  are some residual functions. Upon discretization, the objective function of the discrete time optimal Bolza control problem (5.63) may be expressed as

$$\begin{aligned} f(x, u) &= \sum_{k=0}^{N-1} f_k(x_k, u_k) + f_N(x_N) \\ &= \frac{1}{2} \sum_{k=0}^{N-1} \|\varphi_k(x_k, u_k)\|_{\Lambda_k}^2 + \frac{1}{2} \|\varphi_N(x_N)\|_{\Lambda_N}^2 \end{aligned} \quad (5.200)$$

in which the stage cost functions are

$$f_k(x_k, u_k) = \frac{1}{2} \|\varphi_k(x_k, u_k)\|_{\Lambda_k}^2 \quad k \in \mathcal{N}_0 \quad (5.201a)$$

$$f_N(x_N) = \frac{1}{2} \|\varphi_N(x_N)\|_{\Lambda_N}^2 \quad (5.201b)$$

and  $\mathcal{N}_0 = \{0, 1, \dots, N-1\}$ .  $\varphi_k(x_k, u_k)$  are  $\varphi_N(x_N)$  are discrete-time residual functions obtained from the continuous time residual functions through the particular integration method employed. By construction the residual functions are expected to be small for well-posed problems, as they represent the deviation of some variable from a desired value.

The gradients of the stage costs are

$$\nabla_{x_k} f_k = \nabla_{x_k} \varphi_k(x_k, u_k) \Lambda_k \varphi_k(x_k, u_k) \quad k \in \mathcal{N}_0 \quad (5.202a)$$

$$\nabla_{u_k} f_k = \nabla_{u_k} \varphi_k(x_k, u_k) \Lambda_k \varphi_k(x_k, u_k) \quad k \in \mathcal{N}_0 \quad (5.202b)$$

$$\nabla_{x_N} f_N = \nabla_{x_N} \varphi_N(x_N) \Lambda_N \varphi_N(x_N) \quad (5.202c)$$

From these gradient expressions the second partial derivatives may be deduced

for  $k \in \mathcal{N}_0$

$$\begin{aligned}
\nabla_{x_k, x_k}^2 f_k &= \nabla_{x_k} \varphi_k(x_k, u_k) \Lambda_k \nabla_{x_k} \varphi_k(x_k, u_k)' \\
&\quad + \sum_{i=1}^l [\Lambda_k \varphi_k(x_k, u_k)]_i \nabla_{x_k, x_k}^2 \varphi_{i,k}(x_k, u_k) \\
\nabla_{x_k, u_k}^2 f_k &= \nabla_{x_k} \varphi_k(x_k, u_k) \Lambda_k \nabla_{u_k} \varphi_k(x_k, u_k)' \\
&\quad + \sum_{i=1}^l [\Lambda_k \varphi_k(x_k, u_k)]_i \nabla_{x_k, u_k}^2 \varphi_{i,k}(x_k, u_k) \\
\nabla_{u_k, u_k}^2 f_k &= \nabla_{u_k} \varphi_k(x_k, u_k) \Lambda_k \nabla_{u_k} \varphi_k(x_k, u_k)' \\
&\quad + \sum_{i=1}^l [\Lambda_k \varphi_k(x_k, u_k)]_i \nabla_{u_k, u_k}^2 \varphi_{i,k}(x_k, u_k)
\end{aligned}$$

and

$$\begin{aligned}
\nabla_{x_N, x_N}^2 f_N &= \nabla_{x_N} \varphi_N(x_N) \Lambda_N \nabla_{x_N} \varphi_N(x_N)' \\
&\quad + \sum_{i=1}^l [\Lambda_N \varphi_N(x_N)]_i \nabla_{x_N, x_N}^2 \varphi_{i,N}(x_N)
\end{aligned}$$

One crucial point concerning these expressions near the solution is that the second term is expected to be much smaller than the first term as the residuals are assumed small. Another important feature is that no second order derivatives are needed in evaluation of the first terms. The evaluation of the first terms only requires information about the residual gradients which already have been evaluated in computing the stage cost gradients. Consequently, good approximations of the Hessian of the objective function can be obtained at virtually no computational cost by neglecting the second terms.

The Lagrangian function may as in the non-least squares case be expressed as

$$\begin{aligned}
\mathcal{L}(x, u, \pi) &= \sum_{k=0}^{N-1} f_k(x_k, u_k) + f_N(x_N) \\
&\quad - \sum_{k=0}^{N-1} \pi'_k (H_k(x_k, u_k) - x_{k+1})
\end{aligned}$$

such that its second partial derivatives are given by the expressions

$$\begin{aligned}\nabla_{x_k, x_k}^2 \mathcal{L} &= \nabla_{x_k, x_k}^2 f_k - \sum_{i=1}^m \pi_{i,k} \nabla_{x_k, x_k}^2 H_{i,k}(x_k, u_k) \\ \nabla_{x_k, u_k}^2 \mathcal{L} &= \nabla_{x_k, u_k}^2 f_k - \sum_{i=1}^m \pi_{i,k} \nabla_{x_k, u_k}^2 H_{i,k}(x_k, u_k) \\ \nabla_{u_k, u_k}^2 \mathcal{L} &= \nabla_{u_k, u_k}^2 f_k - \sum_{i=1}^m \pi_{i,k} \nabla_{u_k, u_k}^2 H_{i,k}(x_k, u_k)\end{aligned}$$

and

$$\nabla_{x_N, x_N}^2 \mathcal{L} = \nabla_{x_N, x_N}^2 f_N$$

By the Gauss-Newton assumption, the residuals,  $\varphi_k$  and  $\varphi_N$ , are assumed small relative to the derivatives of the residuals. As has been discussed in Jørgensen *et al.* (2002), at the optimal solution the Lagrange multipliers,  $\pi_k$ , are proportional to the residuals and therefore expected to be small. Consequently, the small residual assumption invoked by the Gauss-Newton approximation implies that the Hessian matrices of the Lagrangian in the least squares problem may be approximated by

$$\begin{aligned}\nabla_{x_k, x_k}^2 \mathcal{L} &\approx \nabla_{x_k} \varphi_k(x_k, u_k) \Lambda_k \nabla_{x_k} \varphi_k(x_k, u_k)' \\ \nabla_{x_k, u_k}^2 \mathcal{L} &\approx \nabla_{x_k} \varphi_k(x_k, u_k) \Lambda_k \nabla_{u_k} \varphi_k(x_k, u_k)' \\ \nabla_{u_k, u_k}^2 \mathcal{L} &\approx \nabla_{u_k} \varphi_k(x_k, u_k) \Lambda_k \nabla_{u_k} \varphi_k(x_k, u_k)'\end{aligned}$$

and

$$\nabla_{x_N, x_N}^2 \mathcal{L} \approx \nabla_{x_N} \varphi_N(x_N) \Lambda_N \nabla_{x_N} \varphi_N(x_N)'$$

By these expressions, it is evident that the Hessians of the Lagrangians in the least squares problem may be approximated using only the residual gradients but not the Hessians of the residuals. The residual gradients are already available as they are used in computation of the gradients  $\nabla_{x_k} f_k(x_k, u_k)$ ,  $\nabla_{u_k} f_k(x_k, u_k)$ , and  $\nabla_{x_N} f_N(x_N)$ . By these expressions, good approximations of the Hessians in the small residual least squares case can be obtained at negligible computational cost.

The introduced approximations of the Lagrangians may be applied for efficient computation of the Hessian matrices used in the SQP algorithms for optimal control. The data related to the objective function used in the SQP algorithms for optimal control may for the small residual least squares problem be

computed by

$$Q_k = \nabla_{x_k} \varphi_k(x_k^0, u_k^0) \Lambda_k \nabla_{x_k} \varphi_k(x_k^0, u_k^0)' \quad (5.205a)$$

$$M_k = \nabla_{x_k} \varphi_k(x_k^0, u_k^0) \Lambda_k \nabla_{u_k} \varphi_k(x_k^0, u_k^0)' \quad (5.205b)$$

$$R_k = \nabla_{u_k} \varphi_k(x_k^0, u_k^0) \Lambda_k \nabla_{u_k} \varphi_k(x_k^0, u_k^0)' \quad (5.205c)$$

$$q_k = \nabla_{x_k} \varphi_k(x_k^0, u_k^0) \Lambda_k \varphi_k(x_k^0, u_k^0) \quad (5.205d)$$

$$r_k = \nabla_{u_k} \varphi_k(x_k^0, u_k^0) \Lambda_k \varphi_k(x_k^0, u_k^0) \quad (5.205e)$$

and

$$P_N = \nabla_{x_N} \varphi_N(x_N^0) \Lambda_N \nabla_{x_N} \varphi_N(x_N^0) \quad (5.206a)$$

$$p_N = \nabla_{x_N} \varphi_N(x_N^0) \Lambda_N \varphi_N(x_N^0) \quad (5.206b)$$

rather than by the modified BFGS approximations.

Essentially, the method described for approximating the Hessian matrix is the Gauss-Newton method for the nonlinear least-squares problem specialized to the least-squares nonlinear optimal control problem. A straightforward improvement of the Hessian matrix obtained by this approach is to apply the refinements of the Gauss-Newton method applied to the general nonlinear least-squares problem. These improvements of the Gauss-Newton method for the general nonlinear least-squares problem are described by Björck (1996) as well as by Seber and Wild (1989). The most successful extensions of the pure Gauss-Newton method use a switching strategy in which the algorithms switch between the Gauss-Newton approximation of the Hessian matrix and the full approximate Hessian matrix. In the NL2SOL algorithm by Dennis *et al.* (1981), the terms neglected by the Gauss-Newton approximation of the Hessian matrix are approximated by a specially tailored symmetric but not necessarily positive definite secant approximation. NL2SOL switches between the model using the Gauss-Newton Hessian approximation and the model using the full Hessian approximation by comparing the predicted objective function reductions with the actual objective function reduction. NL2SOL is implemented in a trust-region framework such that the quadratic program solved at each iteration is positive definite even though the full approximate Hessian matrix is not necessarily positive definite. However, the fact that the full approximate Hessian matrix is not necessarily positive definite makes direct extension of the NL2SOL algorithm to the line-search algorithms for the least-squares nonlinear optimal control problem problematic. Another, extension of the Gauss-Newton method for the general nonlinear least-squares problem has been given by Fletcher and Xu (1987) and is described in Fletcher (1987). This method is embedded in a line-search procedure and switches between a Gauss-Newton approximation of the Hessian matrix and a modified BFGS approximation. As it is a line-search based algorithm, it can be extended to the least-squares nonlinear optimal control problem in a straightforward fashion such that the partitioned BFGS update and the structured Gauss-Newton approximations (5.205)-(5.206) are

applied. For the least-squares nonlinear optimal control problem, Tousain and Bosgra (2000) approximate the terms neglected by the Gauss-Newton approximation using a BFGS approximation and report good results in applying this procedure for nonlinear model predictive control of a polymerization process. This Hessian approximation specialized to the least-squares nonlinear optimal control problem use the same fundamental idea as NL2SOL. However, the main difference is that Tousain and Bosgra (2000) enforce the terms neglected by the Gauss-Newton approximation and approximated by a secant update to be positive definite. Diehl (2001) employ a Gauss-Newton approximation in his solution of the least-squares nonlinear optimal control problem applied for nonlinear model predictive control. Tenny *et al.* (2002) as well as Tenny (2002) discuss several approximations of the Hessian matrix including the Gauss-Newton approximation for the least-squares nonlinear optimal control problem as it arises in nonlinear model predictive control.

#### 5.8.4 Integration Methods

When explicit methods are used for discretization of the continuous time optimal Bolza control problem (5.54), the discrete-time optimal control problem (5.63) arises directly. In contrast, direct application of an implicit integration method for discretization of (5.54) does not lead to a discrete-time problem with the required structure (5.63). In the implicit integration case, the discrete-time objective function and stage cost functions are given by  $f(x, u) = \sum_{k=0}^{N-1} f_k(x_k, u_k, x_{k+1}) + f_N(x_N)$ . Consequently, the structure of the stage costs  $f_k(x_k, u_k, x_{k+1})$  implies that the block diagonal structure of the Hessian of the objective function is lost. Another consequence of this structure is that the corresponding Lagrangian function is no longer partially separable. Similarly, by implicit integration the discrete-time dynamics has the structure  $H_k(x_k, u_k, x_{k+1}) = 0$ . This discrete-time dynamic constraint is not necessarily linear in  $x_{k+1}$  which also implies that the block-diagonal structure of the Hessian of the Lagrangian is lost. These properties of the implicit methods imply that the infeasible path SQP algorithm for optimal control cannot be applied directly in conjunction with this integration method. However, feasible path algorithms are not restricted to a particular integration method and work as well with an implicit integration method. By adopting a multiple shooting approach in which the differential equations are integrated exactly on a number of intervals, the principles of the infeasible path algorithm may be applied as the required discrete-time structure is retained by this procedure (Bock *et al.*, 2000; Binder *et al.*, 2001a).

Explicit integration methods (c.f. Hairer *et al.*, 1993; Hairer and Wanner, 1996) in conjunction with optimal control have been studied by Polak (1997), Schwartz (1996), and Betts (2001) among others. Polak (1997) applied an explicit Euler discretization in his algorithms. Schwartz (1996) applies the same optimization algorithms as Polak (1997) but apply an explicit Runge-Kutta

integration. Betts (2001) present optimal control algorithms based on explicit Euler integration and classic Runge-Kutta integration, but recommend and implement an algorithm based on trapezoidal and Hermite-Simpson integration.

Pytlak (1999) has presented a comprehensive study of optimal control using implicit Runge-Kutta integration. A number of dynamic optimization algorithms have relied on orthogonal collocation for integration of the differential equations (c.f. Biegler, 1984; Cuthrell and Biegler, 1987, 1989; Logsdon and Biegler, 1993; Cervantes *et al.*, 2000) and is typically implemented as part of an infeasible path SQP algorithm (c.f. Binder *et al.*, 2001a). When implicit integrations methods and orthogonal collocation on finite elements are employed in a feasible path SQP algorithm for the optimal control problem, the optimal control problem structure described in this paper is lost. Instead the problem has a general block banded structure, and an appropriate block-banded or a general sparse solver such as MA27 must be employed (c.f. Biegler, 2000; Golub and Van Loan, 1996; Duff *et al.*, 1986).

Multiple shooting algorithms are able to retain the discrete-time optimal control structure of (5.63) in the connection of one shooting interval to the next, while employing standard explicit or implicit sensitivity augmented integration methods (c.f. Bock and Plitt, 1983; Bock *et al.*, 2000). Bauer (2000) has developed a BDF-based integration code used in the software for dynamic optimization by multiple shooting of systems described by either ordinary differential equations or systems of differential algebraic equations. This integration algorithm is employed by Diehl (2001) and applied for nonlinear model predictive control of a distillation column (c.f. Diehl *et al.*, 2001).

The variable step and variable order BDF algorithm has been augmented to provide sensitivity information along with the nominal trajectory by a number of authors. Caracotsios and Stewart (1985) applied a direct staggered method for computation of the sensitivities. Maly and Petzold (1996) computed the sensitivities by a simultaneous corrector method, while Feehery *et al.* (1997) applied a staggered corrector method for computation of the sensitivities. These methods have been implemented by Li and Petzold (1999, 2001) in their augmentation of the popular DASSL algorithm. Serban and Petzold (2000) applied this integration method in a multiple shooting context to optimal control problems. Leis and Kramer (1985), Kiehl (1999), and Lee *et al.* (2000) have also addressed the computation of sensitivities. Galán *et al.* (1999) extended the sensitivity computation to event-based discontinuous differential algebraic equations. High-order integration methods like the BDF may lose efficiency for optimal control problems as the frequent discontinuities due to the piecewise constant parameterization of the optimal control requires frequent restart of the BDF algorithm (c.f. Schlegel and Marquardt, 2001). Schlegel and Marquardt (2001) exploited this observation and demonstrated the efficiency of a linearly-implicit Euler discretization (c.f. Deuffhard, 1983; Deuffhard *et al.*, 1987, 1990) compared to the BDF implementations by Li and Petzold (1999). Due to the frequent discontinuities, the implicit and semi-implicit Runge-



Kutta integration methods may be effective compared to BDF based methods since Runge-Kutta methods are one-step integration methods.

The selection of a proper mesh and in particular a proper mesh for the parameterization of the control profiles such that the discrete optimal control problem (5.63) approximates the continuous-time optimal control problem (5.54) well has been considered by Waldruff *et al.* (1997), Betts and Huffmann (1998), Betts (2001), and Binder *et al.* (2001b). The mesh selection is an extension that is added on top of the solution of the discrete-time optimal control problem.

### 5.8.5 Index One DAEs

Chemical processes operated at a phase equilibrium or a reaction equilibrium are typically described the differential equations and in addition a set of algebraic equations. Systems with such dynamics are called differential algebraic equations (DAE). Application of dynamic optimization algorithms to such processes requires some modifications of the basic algorithm developed for a system of ordinary differential equations. Recently, Leineweber (1999), Diehl (2001), and Leineweber *et al.* (2001) have reported nice comprehensive studies of the dynamic optimization of differential algebraic systems.

An index 1 system of differential algebraic equations is the following system of differential and algebraic equations

$$\dot{x}(t) = h(x(t), z(t), u(t)) \quad (5.207a)$$

$$0 = g(x(t), z(t), u(t)) \quad (5.207b)$$

in which  $\nabla_z g(x, z, u)$  is non-singular.  $x$  is the differential states,  $z$  is the algebraic states, and  $u$  is the process inputs which are available for manipulation. A generic discretization of the system of differential algebraic equations (5.207) may be denoted

$$x_{k+1} = H_k(x_k, z_k, u_k) \quad (5.208a)$$

$$0 = g(x_k, z_k, u_k) \quad (5.208b)$$

Linearization around a nominal trajectory  $\{x_k^0, z_k^0, u_k^0\}_{k=0}^{N-1}$  yields

$$\Delta x_{k+1} = A'_k \Delta x_k + D'_k \Delta z_k + B'_k \Delta u_k + b_k \quad (5.209a)$$

$$0 = E'_k \Delta x_k + F'_k \Delta z_k + G'_k \Delta u_k + d_k \quad (5.209b)$$

in which  $A_k = \nabla_{x_k} H_k(x_k^0, z_k^0, u_k^0)$ ,  $D_k = \nabla_{z_k} H_k(x_k^0, z_k^0, u_k^0)$ ,  $B_k = \nabla_{u_k} H_k(x_k^0, z_k^0, u_k^0)$ ,  $b_k = H_k(x_k^0, z_k^0, u_k^0) - x_{k+1}^0$ ,  $E_k = \nabla_{x_k} g_k(x_k^0, z_k^0, u_k^0)$ ,  $F_k = \nabla_{z_k} g_k(x_k^0, z_k^0, u_k^0)$ ,  $G_k = \nabla_{u_k} g_k(x_k^0, z_k^0, u_k^0)$ , and  $d_k = g_k(x_k^0, z_k^0, u_k^0)$ . As the system is of index 1 by definition, the matrices  $F_k$  in  $\{F_k\}_{k=0}^{N-1}$  are nonsingular. Hence,  $\Delta z_k$  may be expressed by the relation

$$\Delta z_k = -(F'_k)^{-1} E'_k \Delta x_k - (F'_k)^{-1} G'_k \Delta u_k - (F'_k)^{-1} d_k \quad (5.210)$$

Substitution of this expression in the linearized state dynamics yields

$$\begin{aligned}\Delta x_{k+1} &= (A'_k - D'_k(F'_k)^{-1}E'_k)\Delta x_k \\ &\quad + (B'_k - D'_k(F'_k)^{-1}G'_k)\Delta u_k \\ &\quad + (b_k - D'_k(F'_k)^{-1}d_k) \\ &= \tilde{A}'_k\Delta x_k + \tilde{B}'_k\Delta u_k + \tilde{b}_k\end{aligned}\tag{5.211}$$

in which

$$\tilde{A}_k = A_k - E_k F_k^{-1} D_k \tag{5.212a}$$

$$\tilde{B}_k = B_k - G_k F_k^{-1} D_k \tag{5.212b}$$

$$\tilde{b}_k = b_k - (F_k^{-1} D_k)' d_k \tag{5.212c}$$

Consequently, by the projections outlined, the quadratic program resulting from linearization of the differential algebraic optimal control problem has the same staircase structure as the quadratic program resulting from linearization of the optimal control problem with the dynamics described by a system of ordinary differential equations. The Riccati iteration based factorization may be applied for the efficient solution of this quadratic program.

In a number of practical problems the algebraic equations are independent of the process inputs,  $u_k$ , i.e.

$$0 = g_k(x_k, z_k) \tag{5.213}$$

This structure can be exploited to enhance the efficiency of the resulting algorithm even further, as it implies  $G_k = 0$  and  $\tilde{B}_k = B_k$ .

### 5.8.6 Parallelization

Parallelization of the infeasible path sequential quadratic programming algorithm is straightforward as the function evaluations needed in converging the constraints and computing the sensitivities can be conducted independently. This type of relatively coarse parallelization is ideally suited for a heterogeneous cluster of workstations and can be implemented using for instance the message passing interface (MPI) programming standard (c.f. Gropp *et al.*, 1994).

In both the infeasible path methods and the feasible path methods, the sensitivities may be evaluated independently when a nominal trajectory is available. Hence, computational savings by computing the sensitivities in parallel can be expected.

Li and Petzold (1999) have implemented a parallel version of DASSL for solution of differential algebraic systems along with computation of their sensitivities. The parallelization is based on MPI. Similar, the CVODE code has been extended such that it computes sensitivities as well (c.f. Hindmarsh and Serban, 2002) and has been parallelized using MPI (c.f. Lee *et al.*, 2000).

## 5.9 Conclusion

In this paper we have described implementations of infeasible path and feasible path SQP algorithms for unconstrained nonlinear optimal control. The infeasible path method is applicable to solution of systems operated at an unstable equilibrium, while the feasible path method must be of the closed loop type to converge for such a system. The implementation of the closed-loop feasible path SQP algorithm as well as the infeasible path SQP algorithm is facilitated by a structure exploiting Riccati iteration based solver for the sequence of quadratic programs. The SQP based solution procedures are compared qualitatively with the classic solution procedures for unconstrained nonlinear optimal control problems. The implementations and solution of the KKT-systems of the sequence of quadratic programs are solved by either a Riccati iteration based factorization or by solution of a symmetric dense linear system. The Riccati iteration based procedure for solution of the quadratic program enables the closed-loop feasible path algorithm as well as a stabilized implementation of the infeasible path algorithm.

An SQP algorithm for solution of the general equality constrained nonlinear program has been tailored for solution of unconstrained nonlinear optimal control problems. The main ingredients in customization of the general SQP algorithm to the nonlinear optimal control problem are a partitioned modified BFGS update of the approximate Hessian matrix and a Riccati based procedure for solution of the quadratic program. The Riccati based iteration procedure for solution of the quadratic program has been embedded in the SQP algorithms for nonlinear optimal control. This procedure has a computational complexity which is linear in the problem horizon,  $N$ . The alternative to this procedure involves construction of a dense symmetric linear system. Efficient construction of this dense symmetric linear system has been described. A straightforward implementation has a computational complexity which is cubic in the problem horizon,  $N$ . In feasible path SQP algorithms, it is possible to construct the dense system directly and update its Cholesky factors efficiently such that the procedure has a computational complexity which is square in the problem horizon,  $N$ . The developed algorithms constitute the software kernel for nonlinear model predictive control as well as dynamic optimization.

The *principles* for numerical solution of optimal control problems has been illustrated using the unconstrained nonlinear optimal control problem. Based on these principles, extension of the algorithms to constrained nonlinear optimal control problems has been described. Alternatives to the explicit Euler procedure for discretization of the continuous-time optimal control problem have been mentioned as well.

# Conclusion

The extended linear quadratic optimal control problem has been defined and introduced in this thesis. Compared to the linear quadratic regulation problem that is treated in most textbooks, the extended linear quadratic optimal control problem has linear terms in the stage costs and the state transition equation is affine rather than linear.

By combining control theory, optimization theory and dynamic programming efficient solution procedures for the extended linear quadratic optimal control problems with a long horizon has been developed. The solution is based on Riccati recursions. This solution may also be interpreted as the solution of an equality constrained quadratic program. This interpretation provides an efficient method for factorization and solution of the Karush-Kuhn-Tucker necessary and sufficient optimality conditions for the extended linear quadratic optimal control problem. The primal as well the dual variables in the Karush-Kuhn-Tucker optimality conditions are obtained recursively using the Riccati recursions.

The factorization of the Karush-Kuhn-Tucker optimality conditions by the Riccati recursion procedure allows efficient solution of the extended *constrained* linear quadratic programs as well as the nonlinear optimal control problem. The extended constrained linear quadratic optimal control problems may be solved by primal active set methods, dual active set methods, and interior point methods. All these algorithms solve the extended constrained linear quadratic optimal control problem by solving a sequence of extended (unconstrained) linear quadratic optimal control problems. Jørgensen (2004a,c,d) provide details about quadratic programming and the dual active set algorithm. Jørgensen (2004e) shows how a Schur complement technique may be applied for factorization of the KKT-matrix in the dual active set algorithm. This enables application of the dual active set algorithm to the extended constrained linear quadratic optimal control problem, and efficient solution of the problem using Riccati recursions. Jørgensen and Rawlings (2000) show how the Riccati recursions may be applied in interior-point algorithms for solution of the extended linear quadratic optimal control problem.

The Riccati recursion procedure is superior for solution of unstable and ill-conditioned systems compared to alternative feasible path methods. One rea-

son for this is that the Riccati recursion procedure is computationally efficient for long horizons which are usually required for unstable systems. Another reason is that the Riccati recursions generated the optimal feedback control gains as part of the solution. Provided that the system is stabilizable, in a loose sense, the Riccati solution method tends to stabilize the numerical iterations as the optimal feedback gains are computed and applied iteratively.

Assuming that the separation theorem is valid and certainty equivalence holds, it has been argued that the output feedback optimal control problem may be decomposed into an estimation problem and a regulation problem. Both problems are instances of optimal control problems that may be solved efficiently using the solution procedure for the extended linear quadratic optimal control problem.

In addition to facilitate efficient and robust numerical solution of optimal control problems, the extended linear quadratic optimal control problem is also an effective model for formulation of various control problems. In particular, for linear model based predictive control it can readily be used for construction of fast feedforward-feedback predictive controllers (Jørgensen, 2004b), anticipatory predictive control, coordinated decentralized control and processes with active constraints at steady state (Jørgensen and Rawlings, 2000).

The extended linear quadratic optimal control problem may be solved by Riccati recursions. This solution procedure is particularly efficient for problems with long control and prediction horizons. However, for problems with large state dimension, i.e. problems arising from discretization of partial differential equations, the Riccati iteration procedure may be inefficient as the computational cost of the Riccati iterations scales cubically with the state dimensions. For such problems, solution by control vector parameterization may be more efficient.

This thesis has focused on the optimization aspect of the optimal control problem. For nonlinear optimal control problems, the dominating computational cost is solution of the differential equations and computation of the associated sensitivities. Efficient algorithms for this aspect of the problem are provided in Kristensen *et al.* (2004a,b) and Kristensen *et al.* (2005).

# Appendices



# A

## Quadratic Program in SQP

### A.1 Quadratic Program in SQP

This section illustrates how sequential quadratic programming algorithms generate the QP subproblem (Nocedal and Wright, 1999). Consider the nonlinear program

$$\min_{x \in \mathbb{R}^n} f(x) \quad (\text{A.1a})$$

$$s.t. \quad g(x) = 0 \quad (\text{A.1b})$$

$$h(x) \geq 0 \quad (\text{A.1c})$$

in which  $f : \mathbb{R}^n \mapsto \mathbb{R}$ ,  $g$  is a vector function consisting of the functions  $g_i : \mathbb{R}^n \mapsto \mathbb{R}$  for  $i \in \mathcal{E}$ , and  $h$  is a vector function consisting of the functions  $h_i : \mathbb{R}^n \mapsto \mathbb{R}$  for  $i \in \mathcal{I}$ . Let  $\mathcal{A}(x)$  denote the set of active inequality constraints

$$\mathcal{A}(x) = \{i \in \mathcal{I} : h_i(x) = 0\} \quad (\text{A.2})$$

and let  $\mathcal{L}(x, \pi, \lambda)$  be the Lagrangian function of (A.1)

$$\mathcal{L}(x, \pi, \lambda) = f(x) - \pi'g(x) - \lambda'h(x) \quad (\text{A.3})$$

The first order necessary and sufficient optimality conditions for (A.1) are

$$\nabla_x \mathcal{L}(x, \pi, \lambda) = 0 \quad (\text{A.4a})$$

$$g(x) = 0 \quad (\text{A.4b})$$

$$h(x) \geq 0 \quad (\text{A.4c})$$

$$\lambda \geq 0 \quad (\text{A.4d})$$

$$\lambda_i = 0 \quad i \in \mathcal{I} \setminus \mathcal{A}(x) \quad (\text{A.4e})$$

in which

$$\nabla_x \mathcal{L}(x, \pi, \lambda) = \nabla_x f(x) - \nabla_x g(x)\pi - \nabla_x h(x)\lambda \quad (\text{A.5})$$



The search direction in Newtons method for solution of this set of equations must satisfy

$$\begin{aligned} \nabla_{xx}^2 \mathcal{L}(x, \pi, \lambda) \Delta x - \nabla g(x) \Delta \pi - \nabla h(x) \Delta \lambda \\ = -\nabla_x \mathcal{L}(x, \pi, \lambda) \end{aligned} \quad (\text{A.6a})$$

$$\nabla_x g(x)' \Delta x + g(x) = 0 \quad (\text{A.6b})$$

$$\nabla_x h(x)' \Delta x + h(x) \geq 0 \quad (\text{A.6c})$$

$$\Delta \lambda + \lambda \geq 0 \quad (\text{A.6d})$$

$$\Delta \lambda_i + \lambda_i = 0 \quad i \in \mathcal{I} \setminus \tilde{\mathcal{A}}(\Delta x) \quad (\text{A.6e})$$

in which

$$\tilde{\mathcal{A}}(\Delta x) = \{i \in \mathcal{I} : \nabla h_i(x) \Delta x + h_i(x) = 0\} \quad (\text{A.7})$$

Consider the quadratic program

$$\min_{p \in \mathbb{R}^n} \quad \frac{1}{2} p' W p + \nabla f(x)' p \quad (\text{A.8a})$$

$$s.t. \quad \nabla g(x)' p + g(x) = 0 \quad (\text{A.8b})$$

$$\nabla h(x)' p + h(x) \geq 0 \quad (\text{A.8c})$$

The first order necessary and sufficient optimality conditions for this program are

$$Wp + \nabla_x f(x) - \nabla_x g(x)w - \nabla_x h(x)v = 0 \quad (\text{A.9a})$$

$$\nabla_x g(x)' p + g(x) = 0 \quad (\text{A.9b})$$

$$\nabla_x h(x)' p + h(x) \geq 0 \quad (\text{A.9c})$$

$$v \geq 0 \quad (\text{A.9d})$$

$$v_i = 0 \quad i \in \mathcal{I} \setminus \tilde{\mathcal{A}}(p) \quad (\text{A.9e})$$

Let  $W = \nabla_{xx}^2 \mathcal{L}(x, \pi, \lambda)$ . Comparison of (A.6) and (A.9) with this choice of  $W$  yields

$$\begin{bmatrix} \Delta x \\ \Delta \pi \\ \Delta \lambda \end{bmatrix} = \begin{bmatrix} p \\ w - \pi \\ v - \lambda \end{bmatrix} \quad (\text{A.10})$$

Hence, the search direction  $(\Delta x, \Delta \pi, \Delta \lambda)$  of (A.6) may be obtained by solving the quadratic program (A.8).

In line-search SQP methods the next iterate is computed by

$$\begin{bmatrix} x \\ \pi \\ \lambda \end{bmatrix} \leftarrow \begin{bmatrix} x \\ \pi \\ \lambda \end{bmatrix} + \alpha \begin{bmatrix} p \\ w - \pi \\ v - \lambda \end{bmatrix} \quad (\text{A.11})$$

in which  $\alpha \in (0, 1]$  is selected such that some merit function decreases.

### A.1.1 Exact Hessian of Lagrangian

When the search direction is obtained by solution of (A.8) with the Hessian matrix selected as

$$\begin{aligned} W &= \nabla_{xx}^2 \mathcal{L}(x, \pi, \lambda) \\ &= \nabla^2 f(x) - \sum_{i \in \mathcal{E}} \pi_i \nabla^2 g_i(x) - \sum_{i \in \mathcal{I}} \lambda_i \nabla^2 h_i(x) \end{aligned} \quad (\text{A.12})$$

the SQP method may be regarded as a Newton procedure for solution of the conditions (A.6). This choice of  $W$  has two problems. First,  $W$  cannot be guaranteed to be positive definite or positive semi-definite, but may be indefinite. Algorithms do exist for the solution of indefinite quadratic programs (c.f. Gill and Murray, 1978; Gill *et al.*, 1991, 1995). However, they are mainly of the primal active-set type, which are often inferior compared to other algorithms for positive definite programs. The second, and most important objection, concerning this selection of  $W$  is the computational cost of evaluating  $\nabla^2 f(x)$ ,  $\nabla^2 g_i(x)$  for  $i \in \mathcal{E}$ , and  $\nabla^2 h_i(x)$  for  $i \in \mathcal{I}$ .

### A.1.2 Secant Approximation of the Hessian

Instead of solving (A.6) with a Newton method, a quasi-Newton method may be employed for the solution of (A.6). Rather than computing the Jacobian of the non-linear equations (A.6), quasi-Newton methods recur an approximation of the Jacobian which is consistent with the secant equations (c.f. Dennis and Schnabel, 1996; Nocedal and Wright, 1999). The most widely used secant update of the Jacobian matrix for solution of general nonlinear equations is the Broyden update. For unconstrained optimization, the Jacobian matrix of the first order optimality conditions is the Hessian matrix of the objective function. This matrix must be symmetric. The Broyden-Fletcher-Goldfarb-Shanno (BFGS) updates recurs a symmetric positive definite Hessian approximation, which is consistent with the secant equations.

The quasi-Newton solution of (A.6) with a BFGS update of  $\nabla_{xx}^2 \mathcal{L}(x, \pi, \lambda)$ , corresponds to solving the quadratic program (A.8) with the Hessian matrix  $W$  computed as a BFGS-update.

Let  $x_i$  denote the  $i$ th iterate of  $x$  in the SQP algorithm. Then the BFGS approximation of the Hessian  $W$  is computed as

$$W_{i+1} = W_i - \frac{W_i s s' W_i}{s' W_i s} + \frac{y y'}{s' y} \quad (\text{A.13})$$

in which

$$s = x_{i+1} - x_i \quad (\text{A.14a})$$

$$y = \nabla_x \mathcal{L}(x_{i+1}, \pi, \lambda) - \nabla_x \mathcal{L}(x_i, \pi, \lambda) \quad (\text{A.14b})$$

$\pi = \pi_{i+1}$  and  $\lambda = \lambda_{i+1}$  are the most recent computed value of the Lagrange multipliers. The initial approximation of the Hessian matrix,  $W_0$ , may be any positive definite matrix. One option is  $W_0 = I$ .

When  $\nabla_{xx}^2 \mathcal{L}$  is indefinite, the BFGS update may face numerical troubles. To overcome these, it is standard practice to employ a damped BFGS update

$$W_{i+1} = W_i - \frac{W_i s s' W_i}{s' W_i s} + \frac{r r'}{s' r} \quad (\text{A.15})$$

in which

$$r = \theta y + (1 - \theta) W_i s \quad (\text{A.16})$$

with

$$\theta = \begin{cases} 1 & s' y \geq 0.2 s' W_i s \\ \frac{0.8 s' W_i s}{s' W_i s - s' y} & s' y < 0.2 s' W_i s \end{cases} \quad (\text{A.17})$$

These modifications assure that the update is always well-defined. Efficient techniques to update the Cholesky factor of  $W_{i+1}$  rather than  $W_{i+1}$  itself do exist (c.f. Dennis and Schnabel, 1996).

### A.1.3 Gauss-Newton Approximation in Least-Squares Problems

For inverse problems such as the control and state estimation problem (c.f. Goodwin, 2002), general least squares objective functions constitute an important special case whose structure may be utilized in computing an approximation of the Hessian matrix  $W$  for (A.8) (c.f. Dennis and Schnabel, 1996; Bock, 1983).

The nonlinear least squares objective function is

$$f(x) = \frac{1}{2} \|\psi(x)\|_2^2 = \frac{1}{2} \psi(x)' \psi(x) \quad (\text{A.18})$$

in which  $\psi : \mathbb{R}^n \mapsto \mathbb{R}^q$  is a residual function. Such a function is often small at the optimal solution.

The gradient  $\nabla f(x)$  and the Hessian  $\nabla^2 f(x)$  of the nonlinear least squares objective function are

$$\nabla f(x) = \sum_{i=1}^q \psi_i(x) \nabla \psi_i(x) = \nabla \psi(x) \psi(x) \quad (\text{A.19})$$

$$\nabla^2 f(x) = \nabla \psi(x) \nabla \psi(x)' + \sum_{i=1}^q \psi_i(x) \nabla^2 \psi_i(x) \quad (\text{A.20})$$

in which we use the following convention for the gradient of vector functions

$$\nabla\psi = \begin{bmatrix} \frac{\partial\psi_1}{\partial x_1} & \cdots & \frac{\partial\psi_q}{\partial x_1} \\ \vdots & & \vdots \\ \frac{\partial\psi_1}{\partial x_n} & \cdots & \frac{\partial\psi_q}{\partial x_n} \end{bmatrix} \quad (\text{A.21})$$

Assuming that the residual  $\psi(x)$  are small, the second term of (A.20) becomes negligible compared to the first term. Hence the Hessian matrix  $\nabla^2 f(x)$  may be approximated by

$$\nabla^2 f(x) \approx \nabla\psi(x)\nabla\psi(x)' \quad (\text{A.22})$$

The computation of this approximation is cheap, as  $\nabla\psi(x)$  is already available as it is also used in the computation of  $\nabla f(x)$ . By construction, this approximation provides a symmetric positive semi-definite Hessian. For most practical problems  $\psi(x)$  is chosen such that the Hessian matrix in the approximation (A.22) is even symmetric positive definite.

The preceding analysis is meant to facilitate the analysis of the weighted nonlinear least squares problem

$$f(x) = \frac{1}{2}\|\varphi(x)\|_Q^2 = \frac{1}{2}\varphi(x)'Q\varphi(x) \quad (\text{A.23})$$

in which  $\varphi : \mathbb{R}^n \mapsto \mathbb{R}^q$  is a residual function and  $Q \in \mathbb{R}^{q \times q}$  is symmetric positive definite. The weighted nonlinear least squares problem (A.23) may be regarded as a particular instance of the nonlinear least squares problem (A.18)

$$f(x) = \frac{1}{2}\|\varphi(x)\|_Q^2 = \frac{1}{2}\|L\varphi(x)\|_2^2 = \frac{1}{2}\|\psi(x)\|_2^2 \quad (\text{A.24})$$

in which  $L \in \mathbb{R}^{q \times q}$  is the Cholesky factor of  $Q$ , i.e.  $Q = LL' = L'L$ , and  $\psi(x) = L\varphi(x)$ . Hence,  $\nabla\psi(x) = \nabla\varphi(x)L'$  implies that the gradient  $\nabla f(x)$  is

$$\begin{aligned} \nabla f(x) &= \nabla\psi(x)\psi(x) \\ &= \nabla\varphi(x)L'L\varphi(x) = \nabla\varphi(x)Q\varphi(x) \end{aligned} \quad (\text{A.25})$$

and the Hessian matrix  $\nabla^2 f(x)$  may be approximated by

$$\begin{aligned} \nabla^2 f(x) &\approx \nabla\psi(x)\nabla\psi(x)' \\ &= \nabla\varphi(x)L'L\nabla\varphi(x)' = \nabla\varphi(x)Q\nabla\varphi(x)' \end{aligned} \quad (\text{A.26})$$

Consider the nonlinear program (A.1) with the objective function (A.23). For simplicity of the argument but without loss of generality, assume that the inequality constraint are inactive. This implies  $\lambda = 0$  and condition (A.4a) becomes

$$\begin{aligned} \nabla_x \mathcal{L}(x, \pi, \lambda) &= \nabla f(x) - \nabla g(x)\pi - \nabla h(x)\lambda \\ &= \nabla\varphi(x)Q\varphi(x) - \nabla g(x)\pi = 0 \end{aligned} \quad (\text{A.27})$$

Hence,  $\pi$  may be regarded as the following function

$$\pi = [\nabla g(x)]^\dagger \nabla \varphi(x) Q \varphi(x) \quad (\text{A.28})$$

in which  $[\nabla g(x)]^\dagger$  is the pseudo-inverse of  $\nabla g(x)$ .  $\pi$  is thus proportional to the residual function  $\varphi(x)$  and expected to be small. Therefore, the Hessian of the Lagrangian,  $\nabla_{xx}^2 \mathcal{L}$ , may be approximated according to

$$\begin{aligned} \nabla_{xx}^2 \mathcal{L}(x, \pi, \lambda) &= \nabla^2 f(x) - \sum_{i \in \mathcal{E}} \pi_i \nabla^2 g_i(x) - \sum_{i \in \mathcal{I}} \lambda_i \nabla^2 h_i(x) \\ &\approx \nabla \varphi(x) Q \nabla \varphi(x)' \end{aligned} \quad (\text{A.29})$$

Consequently, under the small residual assumption, the quadratic program (A.8) for computing the search direction in an SQP iteration of the nonlinear program (A.1) with a weighted nonlinear least squares objective function (A.22) may be generated with  $\nabla f(x)$  computed by (A.25) and the Hessian matrix,  $W$ , computed by

$$W = \nabla \varphi(x) Q \nabla \varphi(x)' \quad (\text{A.30})$$

#### A.1.4 Hybrid Methods for Least-Squares Problems

In the large residual case methods based on the Gauss-Newton assumption may fail and one may have to resort to hybrid methods.

One method, whose idea is due to Fletcher and Xu (1987) maintains a sequence of positive definite Hessian approximations,  $W_i$ . At each iteration a Gauss-Newton step is attempted and accepted if it reduced the merit function by a certain factor. In case of acceptance, the Hessian approximation,  $W_i$ , is overwritten with the Gauss-Newton approximation. Otherwise, a new direction is obtained using  $W_i$ , and a new point is obtained by doing a line search. In either case, a BFGS-like update is applied to  $W_i$  to obtain the next approximation  $W_{i+1}$ . This method is well suited to be adapted in line-search SQP algorithms.

Dennis *et al.* (1981) employ a different method in their hybrid code, NL2SOL. Basically, they use a secant update to approximate the part neglected by the Gauss-Newton approximation. They then use this update and the Gauss-Newton approximation in a trust region method. Due to the technical details of this method it is not well suited for line search SQP algorithms.

# B

## Dense MPC Quadratic Program

The subject of this chapter is the transformation of a model predictive control quadratic program into a standard quadratic program. The utility of this result is that standard solvers for quadratic programming can be applied in the computation of the optimal values for the actuators defined by the MPC quadratic program.

Model predictive control is introduced in a process control context by Ogunnaike and Ray (1994) and Seborg *et al.* (2004). Camacho and Bordons (1999) and Maciejowski (2002) provide a text book description of model predictive control. Garcia *et al.* (1989), Qin and Badgwell (1996, 2000), Morari and Lee (1999), Allgöwer *et al.* (1999), Rawlings (2000), and Mayne *et al.* (2000) provide surveys of various aspects of predictive control. Quadratic programs for solution of model predictive control problems are discussed by Wright (1996), Rao *et al.* (1998), Bartlett *et al.* (2000), Bartlett *et al.* (2002), Tiagounov *et al.* (2003). However, these works do not present detailed descriptions about the construction of a dense QP for the predictive control problem. This chapter provides such a description and mention standard QP solvers by which the resulting quadratic programs may be solved. The presentation is limited to formulation of dense quadratic programs for constrained extended linear quadratic regulators.

### B.1 Standard Quadratic Program

In this section the standard quadratic program is defined. The standard quadratic program is in a form which is readily solvable by general solvers for quadratic programming. Furthermore in this context of MPC we choose a standard form which is particularly relevant for this special case. For other purposes such as algorithmic construction for quadratic programs other forms depending on the particular method are more convenient.

**Definition B.1.1 (Standard Quadratic Program)**

Let  $Q \in \mathbb{R}^{n \times n}$  and positive definite,  $q \in \mathbb{R}^n$ ,  $\rho \in \mathbb{R}$ ,  $A \in \mathbb{R}^{m \times n}$ ,  $l \in \mathbb{R}^m$ ,  $u \in \mathbb{R}^m$ ,  $\pi_{min} \in \mathbb{R}^n$ , and  $\pi_{max} \in \mathbb{R}^n$  be given data defining the standard quadratic program:

$$\min_{\pi} \quad \phi = \frac{1}{2} \pi' Q \pi + q' \pi + \rho \quad (\text{B.1a})$$

$$s.t. \quad l \leq A\pi \leq u \quad (\text{B.1b})$$

$$\pi_{min} \leq \pi \leq \pi_{max} \quad (\text{B.1c})$$

The optimal solution is denoted  $\pi^*$  provided it exists. Similar the optimal value of the objective function is  $\phi^*$ .

**Remark B.1.2**

The constant term,  $\rho$ , in the objective function does not affect the solution,  $\pi^*$ . It only affects the value of the objective function. Hence, it can be discarded in definition of the standard quadratic program, if only the solution  $\pi^*$  and not the actual value of the objective function is of interest.

**Remark B.1.3**

The standard quadratic program may have no solution because the feasible set is empty. Quality implementations for quadratic programming identify these situations and terminate with appropriate information about the reason of termination. Unboundedness is not an issue as  $Q$  is positive definite.

**Remark B.1.4**

The positive definiteness of  $Q$  ensures that the solution to the standard quadratic program is unique provided it exists. It also means that the quadratic program is strictly convex.

**Remark B.1.5**

The solution,  $\pi^*$  may be regarded as a function of the data, i.e.

$$\begin{aligned} \pi^* &= \mu(Q, q, A, l, u, \pi_{min}, \pi_{max}) \\ &= \arg \min_{\pi} \{ \phi : l \leq A\pi \leq u, \pi_{min} \leq \pi \leq \pi_{max} \} \end{aligned} \quad (\text{B.2})$$

The standard quadratic program is identical with the form used by Gill *et al.* (1995) in QPSOL. The QPSOL algorithm is described in Gill and Murray (1978) and Gill *et al.* (1991) and is a primal active-set algorithm.

The other major active-set method is a dual method due to Goldfarb and Idnani (1983). This algorithm is further discussed and enhanced by Powell (1985a) and Schmid and Biegler (1994). The implementation by Powell (1985a) is available in the IMSL library as QPROG/DQPROG. This implementation is also available in the Harwell library (c.f. AEA Technology, 1995) as VE07 and VE17 as well as in an implementation due to Schittkowski (2002, 2003). The quadratic programming algorithm due to Goldfarb and Idnani (1983) applies to strictly convex problems only, while the method by Gill *et al.* (1995) even applies to indefinite quadratic programs. However, reasonable formulated model predictive control problems are convex, so the restriction to strictly convex quadratic

programs is no limitation with regard to solution of model predictive control quadratic programs.

A third class of algorithms for solution of dense quadratic programs is interior point methods. These methods are particularly simple to implement and are excellently described in Wright (1997) and Nocedal and Wright (1999). Interior point methods are generally believed to be more effective for structured problems in which the structure can be effectively applied in solving the appropriate linear equations. Gertz and Wright (2001a,b) provide an object oriented C++ implementation of interior point algorithms for quadratic programming. One shortcoming of these infeasible path interior point methods is that they cannot robustly detect infeasibility. By using the homogeneous and self-dual formulation, numerically efficient primal-dual interior-point algorithms for convex quadratic programming capable of detecting infeasibility robustly have been constructed (Andersen, 2000; Andersen and Andersen, 2000; Andersen *et al.*, 2003; Andersen and Ye, 1998, 1999). These algorithms are implemented in MOSEK<sup>1</sup>.

## B.2 LTI MPC Quadratic Program

A practical way to solve the quadratic program defining the model predictive controller is to transform this quadratic program into a standard quadratic program and apply a readily available solver for the standard quadratic program. Hence, first we define a quadratic program defining a class model predictive controllers with linear time invariant models, and next we transform this into a standard quadratic program.

### Definition B.2.1 (LTI MPC Quadratic Program)

The quadratic program defining the model predictive controller with a linear time invariant model is

$$\min_{\{x_{k+1}, u_k\}_{k=0}^{N-1}} \phi = \frac{1}{2} \sum_{k=0}^{N-1} \{x'_k Q x_k + 2x'_k M u_k + u'_k R u_k + 2q'_k x_k + 2r'_k u_k\} \quad (\text{B.3a})$$

$$+ \frac{1}{2} x'_N P x_N + p' x_N \quad (\text{B.3b})$$

$$s.t. \quad x_{k+1} = A x_k + B u_k \quad k = 0, 1, \dots, N-1 \quad (\text{B.3c})$$

$$u_{min} \leq u_k \leq u_{max} \quad k = 0, 1, \dots, N-1 \quad (\text{B.3d})$$

$$b_l \leq G x_k + H u_k \leq b_u \quad k = 0, 1, \dots, N-1 \quad (\text{B.3e})$$

$$\tilde{b}_l \leq \tilde{G} x_N \leq \tilde{b}_u \quad (\text{B.3f})$$

in which the data is  $x_0, \{q_k\}_{k=0}^{N-1}, \{r_k\}_{k=0}^{N-1}, p, Q, M, R, P, A, B, G, H, b_l, b_u, \tilde{G}, \tilde{b}_l, \tilde{b}_u, u_{min}$ , and  $u_{max}$ .

---

<sup>1</sup>[www.mosek.com](http://www.mosek.com)



The solution is both the control sequence  $\{u_k^*\}_{k=0}^{N-1}$  and the state sequence  $\{x_k^*\}_{k=1}^N$ . The corresponding optimal value of the objective function is denoted  $\phi^*$ .

**Remark B.2.2**

The optimal control sequence,  $\{u_k^*\}$ , can be regarded as the following function

$$\{u_k^*\}_{k=0}^{N-1} = \mu(x_0, \{q_k\}_{k=0}^{N-1}, \{r_k\}_{k=0}^{N-1}, p; Q, M, R, P, A, B, G, H, b_l, b_u, \tilde{G}, \tilde{b}_l, \tilde{b}_u, u_{min}, u_{max}) \quad (\text{B.4})$$

in which the first part of the input arguments may possibly change online. The second part of the input arguments may be regarded as design parameters and therefore regarded as constants.

**Definition B.2.3 (Augmented Transition Matrix)**

The augmented transition matrix is defined as

$$\Phi = \begin{pmatrix} A \\ A^2 \\ \vdots \\ A^N \end{pmatrix} \quad (\text{B.5})$$

**Definition B.2.4 (Reverse Augmented Controllability Matrix)**

$$\Gamma = \begin{pmatrix} \Gamma_1 \\ \Gamma_2 \\ \Gamma_3 \\ \vdots \\ \Gamma_{N-1} \\ \Gamma_N \end{pmatrix} = \begin{pmatrix} B & 0 & 0 & \dots & 0 & 0 \\ AB & B & 0 & \dots & 0 & 0 \\ A^2B & AB & B & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ A^{N-2}B & A^{N-3}B & A^{N-4}B & \dots & B & 0 \\ A^{N-1}B & A^{N-2}B & A^{N-3}B & \dots & AB & B \end{pmatrix} \quad (\text{B.6})$$

**Remark B.2.5**

$\Gamma_N$  is

$$\Gamma_N = (A^{N-1}B \quad A^{N-2}B \quad A^{N-3}B \quad \dots \quad AB \quad B) \quad (\text{B.7})$$

and  $\Gamma_k$  ( $k < N$ ) is directly available from  $\Gamma_N$  by copying the last  $k$  block-columns of  $\Gamma_N$ . No re-computation is needed.

**Definition B.2.6 (Vector of Control and States)**

Define the vector of controls,  $\pi$ , and the vector of states,  $x$ , as

$$\pi = \begin{pmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{pmatrix} \quad x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} \quad (\text{B.8})$$

**Lemma B.2.7 (Expression for the State Dynamics)**

The state dynamics  $x_{k+1} = Ax_k + Bu_k$   $k = 0, 1, 2, \dots$  implies that the states of the system may be expressed in terms of the control and the initial state according to

$$\begin{aligned} x_k &= A^k x_0 + \sum_{j=0}^{k-1} A^{k-1-j} Bu_j \\ &= A^k x_0 + \Gamma_k \pi \end{aligned} \quad k = 1, 2, \dots \quad (\text{B.9})$$

Using the notation introduced by the above definitions, this expression for  $k = 1, \dots, N$  may be expressed as

$$x = \Phi x_0 + \Gamma \pi \quad (\text{B.10})$$

*Proof.* The first part concerns proving that the state dynamics  $x_{k+1} = Ax_k + Bu_k$  implies  $x_k = A^k x_0 + \sum_{j=0}^{k-1} A^{k-1-j} Bu_j$ . The proof is by recursion. Note that the statement

$$P(k) : \quad x_k = A^k x_0 + \sum_{j=0}^{k-1} A^{k-1-j} Bu_j \quad (\text{B.11})$$

is true for  $k = 1$ , i.e.

$$P(k=1) : \quad x_1 = Ax_0 + Bu_0 = A^1 x_0 + \sum_{j=0}^{1-1} A^{1-1-j} Bu_j \quad (\text{B.12})$$

Next assume that the statement  $P(k)$  is true. We want to show that this implies that  $P(k+1)$  is also true. For convenience let  $l = k+1$ . The system dynamics  $x_{k+1} = Ax_k + Bu_k$  and  $P(k)$  implies

$$\begin{aligned} x_l &= x_{k+1} = Ax_k + Bu_k \\ &= A \left( A^k x_0 + \sum_{j=0}^{k-1} A^{k-1-j} Bu_j \right) + Bu_k \\ &= A^{k+1} x_0 + \sum_{j=0}^{k-1} A^{(k+1)-1-j} Bu_j + A^{(k+1)-1-k} Bu_k \\ &= A^{k+1} x_0 + \sum_{j=0}^k A^{(k+1)-1-j} Bu_j \\ &= A^l x_0 + \sum_{j=0}^{l-1} A^{l-1-j} Bu_j \end{aligned} \quad (\text{B.13})$$

Hence,  $P(l) = P(k+1)$  is true whenever  $P(k)$  is true. Since  $P(1)$  is true and  $P(k) \implies P(k+1)$ ,  $P(k)$  is true for all  $k \in \{1, 2, \dots\}$ . It should be noted that  $P(0)$  is also true.

Secondly, we show  $x_k = A^k x_0 + \Gamma_k \pi$ . This proof is accomplished by simple algebraic manipulations according to

$$\begin{aligned}
 x_k &= A^k x_0 + \sum_{j=0}^{k-1} A^{k-1-j} B u_j \\
 &= A^k x_0 + \begin{pmatrix} A^{k-1}B & A^{k-2}B & A^{k-3}B & \dots & AB & B & 0 & \dots & 0 \end{pmatrix} \begin{pmatrix} u_0 \\ u_1 \\ u_2 \\ \vdots \\ u_{k-2} \\ u_{k-1} \\ u_k \\ \vdots \\ u_{N-1} \end{pmatrix} \\
 &= A^k x_0 + \Gamma_k \pi
 \end{aligned} \tag{B.14}$$

The last part of the lemma is obtained by vectorization of the above result, i.e.

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} = \begin{pmatrix} A \\ A^2 \\ \vdots \\ A^N \end{pmatrix} x_0 + \begin{pmatrix} \Gamma_1 \\ \Gamma_2 \\ \vdots \\ \Gamma_N \end{pmatrix} \pi \tag{B.15}$$

Hence,  $x = \Phi x_0 + \Gamma \pi$ .  $\square$

Lemma B.2.7 is central for the expression of the model predictive control quadratic program as a dense quadratic program, because it is used for elimination of the state dynamics and the states in the model predictive controller quadratic program.

The following very simple lemma is used for transforming the bounds on the actuators to the standard formulation in terms of  $\pi$ .

**Lemma B.2.8 (Actuator Constraints)**

The actuator constraints  $u_{min} \leq u_k \leq u_{max}$  for  $k = 0, 1, \dots, N-1$  may be expressed as

$$\pi_{min} \leq \pi \leq \pi_{max} \tag{B.16}$$

in which

$$\pi_{min} = \begin{pmatrix} u_{min} \\ u_{min} \\ \vdots \\ u_{min} \end{pmatrix} \quad \pi_{max} = \begin{pmatrix} u_{max} \\ u_{max} \\ \vdots \\ u_{max} \end{pmatrix}$$

*Proof.* The result is straightforward and obtained by vectorization of  $u_{min} \leq u_k \leq u_{max}$  for  $k = 0, 1, \dots, N-1$ .  $\square$

Transformation of the general constraints in the model predictive quadratic program is accomplished with the result in the lemma below. The central part of this result is the elimination of the states by application of lemma B.2.7.

**Lemma B.2.9 (General Constraints)**

The general constraints in the model predictive controller quadratic program

$$b_l \leq Gx_k + Hu_k \leq b_u \quad k = 0, 1, \dots, N-1 \quad (\text{B.17a})$$

$$\tilde{b}_l \leq \tilde{G}x_N \leq \tilde{b}_u \quad (\text{B.17b})$$

may be expressed in terms of  $x_0$  and  $\pi$  by

$$b_l - Gx_0 \leq HI_0\pi \leq b_u - Gx_0 \quad (\text{B.18a})$$

$$b_l - GA^k x_0 \leq (G\Gamma_k + HI_k)\pi \leq b_u - GA^k x_0 \quad k = 1, 2, \dots, N-1 \quad (\text{B.18b})$$

$$\tilde{b}_l - \tilde{G}A^N x_0 \leq \tilde{G}\Gamma_N \pi \leq \tilde{b}_u - \tilde{G}A^N x_0 \quad (\text{B.18c})$$

*Proof.*  $u_k$  is eliminated from the expressions by the substitution  $u_k = I_k \pi$ . The situation for  $k = 0$  is obtained by trivial rearrangement of terms. For  $k > 0$  the system dynamics  $x_k = A^k x_0 + \Gamma_k \pi$  is applied and the result is obtained by substitution of this expression for  $x_k$  and subsequent rearrangement of terms.  $\square$

**Remark B.2.10**

$I_k$  for  $k = 0, 1, \dots, N-1$  is defined as

$$I_0 = \begin{bmatrix} I & 0 & 0 & \dots & 0 \end{bmatrix} \quad (\text{B.19a})$$

$$I_1 = \begin{bmatrix} 0 & I & 0 & \dots & 0 \end{bmatrix} \quad (\text{B.19b})$$

$$I_2 = \begin{bmatrix} 0 & 0 & I & \dots & 0 \end{bmatrix} \quad (\text{B.19c})$$

$$\vdots$$

$$I_{N-1} = \begin{bmatrix} 0 & 0 & 0 & \dots & I \end{bmatrix} \quad (\text{B.19d})$$

The general constraints may be expressed compactly according to the following lemma, which is directly compatible with the general constraints in the standard quadratic program.

**Lemma B.2.11 (General Constraints)**

The general inequality constraints may be expressed by

$$l \leq A\pi \leq u \quad (\text{B.20})$$

in which the coefficient matrix  $A$  is

$$A \leftarrow \begin{pmatrix} H & 0 & 0 & \dots & 0 \\ GB & H & 0 & \dots & 0 \\ GAB & GB & H & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ GA^{N-2}B & GA^{N-3}B & GA^{N-4}B & \dots & H \\ \tilde{G}A^{N-1}B & \tilde{G}A^{N-2}B & \tilde{G}A^{N-3}B & \dots & \tilde{G}B \end{pmatrix} \quad (\text{B.21})$$

and the lower and upper bounds are

$$l \leftarrow \begin{pmatrix} b_l \\ b_l \\ b_l \\ \vdots \\ b_l \\ \tilde{b}_l \end{pmatrix} - \begin{pmatrix} G \\ GA \\ GA^2 \\ \vdots \\ GA^{N-1} \\ \tilde{G}A^N \end{pmatrix} x_0 \quad u \leftarrow \begin{pmatrix} b_u \\ b_u \\ b_u \\ \vdots \\ b_u \\ \tilde{b}_u \end{pmatrix} - \begin{pmatrix} G \\ GA \\ GA^2 \\ \vdots \\ GA^{N-1} \\ \tilde{G}A^N \end{pmatrix} x_0 \quad (\text{B.22})$$

*Proof.* The results follow directly by stacking of the expressions in lemma B.2.9.  $\square$

**Remark B.2.12**

The explicit expression for the system dynamics is

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} = \begin{pmatrix} A \\ A^2 \\ \vdots \\ A^N \end{pmatrix} x_0 + \begin{pmatrix} B & 0 & \dots & 0 \\ AB & B & \dots & 0 \\ \vdots & \vdots & & \vdots \\ A^{N-1}B & A^{N-2}B & \dots & B \end{pmatrix} \begin{pmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{pmatrix} \quad (\text{B.23})$$

while the expression for the general constraints may be written as

$$\begin{pmatrix} b_l \\ b_l \\ b_l \\ \vdots \\ b_l \\ \tilde{b}_l \end{pmatrix} \leq \begin{pmatrix} G \\ GA \\ GA^2 \\ \vdots \\ GA^{N-1} \\ \tilde{G}A^N \end{pmatrix} x_0 + \begin{pmatrix} H & 0 & 0 & \dots & 0 & 0 \\ GB & H & 0 & \dots & 0 & 0 \\ GAB & GB & H & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ GA^{N-2}B & GA^{N-3}B & GA^{N-4}B & \dots & GB & H \\ \tilde{G}A^{N-1}B & \tilde{G}A^{N-2}B & \tilde{G}A^{N-3}B & \dots & \tilde{G}AB & \tilde{G}B \end{pmatrix} \begin{pmatrix} u_0 \\ u_1 \\ u_2 \\ \vdots \\ u_{N-2} \\ u_{N-1} \end{pmatrix} \leq \begin{pmatrix} b_u \\ b_u \\ b_u \\ \vdots \\ b_u \\ \tilde{b}_u \end{pmatrix} \quad (\text{B.24})$$

and the expression for the actuator constraints is

$$\begin{pmatrix} u_{min} \\ u_{min} \\ \vdots \\ u_{min} \end{pmatrix} \leq \begin{pmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{pmatrix} \leq \begin{pmatrix} u_{max} \\ u_{max} \\ \vdots \\ u_{max} \end{pmatrix} \quad (\text{B.25})$$

The final major step is to obtain an expression for the objective functions in terms of the actuator variables,  $\pi$ , and the initial state,  $x_0$ . Application of lemma B.2.7 to the objective function yields the desired result. This result is stated in the lemma below.

**Lemma B.2.13 (Dense MPC QP Objective Function)**

The objective function of the model predictive quadratic program (B.3a) may be expressed as

$$\phi = \frac{1}{2} \pi' Q \pi + q' \pi + \rho \quad (\text{B.26})$$

in which

$$\rho \leftarrow \frac{1}{2} x_0' \left( \sum_{k=0}^{N-1} (A^k)' Q A^k + (A^N)' P A^N \right) x_0 + \left( \sum_{k=0}^{N-1} (A^k)' q_k + (A^N)' p \right)' x_0 \quad (\text{B.27a})$$

$$q \leftarrow \left( \sum_{k=0}^{N-1} I_k' M' A^k + \sum_{k=1}^{N-1} \Gamma_k' Q A^k + \Gamma_N' P A^N \right) x_0 + \sum_{k=0}^{N-1} I_k' r_k + \sum_{k=1}^{N-1} \Gamma_k' q_k + \Gamma_N' p \quad (\text{B.27b})$$

$$Q \leftarrow \sum_{k=0}^{N-1} I_k' R I_k + \sum_{k=1}^{N-1} \Gamma_k' Q \Gamma_k + \Gamma_k' M I_k + I_k' M' \Gamma_k + \Gamma_N' P \Gamma_N \quad (\text{B.27c})$$

*Proof.* The proof proceeds by straightforward algebraic manipulations of each term in the objective function

$$\phi = \frac{1}{2} \sum_{k=0}^{N-1} (x_k' Q x_k + 2x_k' M u_k + u_k' R u_k + 2q_k' x_k + 2r_k' u_k) + \frac{1}{2} x_N' P x_N + p' x_N \quad (\text{B.28})$$

Each term involving  $x_k$  is transformed to an expression in  $x_0$  and  $\pi$  using the equation

$$x_k = A^k x_0 + \Gamma_k \pi \quad k = 1, 2, \dots, N \quad (\text{B.29})$$

The quadratic term in  $x_k$  becomes ( $k = 1, 2, \dots, N-1$ )

$$\begin{aligned} x_k' Q x_k &= (A^k x_0 + \Gamma_k \pi)' Q (A^k x_0 + \Gamma_k \pi) \\ &= x_0' (A^k)' Q A^k x_0 + 2x_0' (A^k)' Q \Gamma_k \pi + \pi' \Gamma_k' Q \Gamma_k \pi \end{aligned} \quad (\text{B.30})$$

and the linear term in  $x_k$  may be expressed as ( $k = 1, 2, \dots, N-1$ )

$$q_k' x_k = q_k' (A^k x_0 + \Gamma_k \pi) = q_k' A^k x_0 + q_k' \Gamma_k \pi \quad (\text{B.31})$$

Similarly, the quadratic term at  $k = N$  become

$$\begin{aligned} x'_N P x_N &= (A^N x_0 + \Gamma_N \pi)' P (A^N x_0 + \Gamma_N \pi) \\ &= x'_0 (A^N)' P A^N x_0 + 2x'_0 (A^N)' P \Gamma_N \pi + \pi' \Gamma'_N P \Gamma_N \pi \end{aligned} \quad (\text{B.32})$$

and the linear term in  $x$  at  $k = N$  is

$$p' x_N = p' (A^N x_0 + \Gamma_N \pi) = p' A^N x_0 + p' \Gamma_N \pi \quad (\text{B.33})$$

The mixed term in  $x$  and  $u$  is

$$x'_k M u_k = (A^k x_0 + \Gamma_k \pi)' M I_k \pi = x'_0 (A^k)' M I_k \pi + \pi' \Gamma'_k M I_k \pi \quad (\text{B.34})$$

Finally, the quadratic term in  $u$  is ( $k = 0, 1, \dots, N$ )

$$u'_k R u_k = \pi' I'_k R I_k \pi \quad (\text{B.35})$$

and the linear term in  $u$  is expressed as

$$r'_k u_k = r'_k I_k \pi \quad (\text{B.36})$$

The objective function in the model predictive quadratic program may be utilization of the linear dynamics be expressed as a function explicitly in  $\pi$  and

through the parameters implicitly in  $x_0$

$$\begin{aligned}
\phi &= \frac{1}{2} \sum_{k=0}^{N-1} (x'_k Q x_k + 2x'_k M u_k + u'_k R u_k + 2q'_k x_k + 2r'_k u_k) + \frac{1}{2} x'_N P x_N + p' x_N \\
&= \frac{1}{2} (x'_0 Q x_0 + 2x'_0 M I_0 \pi + \pi' I'_0 R I_0 \pi + 2q'_0 x_0 + 2r'_0 I_0 \pi) \\
&\quad + \frac{1}{2} \sum_{k=1}^{N-1} x'_0 (A^k)' Q A^k x_0 + 2x'_0 (A^k)' Q \Gamma_k \pi + \pi' \Gamma'_k Q \Gamma_k \pi \\
&\quad \quad + 2x'_0 (A^k)' M I_k \pi + \pi' (\Gamma'_k M I_k + I'_k M' \Gamma_k) \pi \\
&\quad \quad + \pi' I'_k R I_k \pi \\
&\quad \quad + 2(q'_k A^k x_0 + q'_k \Gamma_k \pi) + 2r'_k I_k \pi \\
&\quad + \frac{1}{2} (x'_0 (A^N)' P A^N x_0 + 2x'_0 (A^N)' P \Gamma_N \pi + \pi' \Gamma'_N P \Gamma_N \pi) + p' A^N x_0 + p' \Gamma_N \pi \\
&= \frac{1}{2} x'_0 \left( Q + \sum_{k=1}^{N-1} (A^k)' Q A^k + (A^N)' P A^N \right) x_0 + \left( q_0 + \sum_{k=1}^{N-1} (A^k)' q_k + (A^N)' p \right)' x_0 \\
&\quad + \left[ x'_0 \left( M I_0 + \sum_{k=1}^{N-1} (A^k)' Q \Gamma_k + (A^k)' M I_k + (A^N)' P \Gamma_N \right) \right. \\
&\quad \quad \left. + r'_0 I_0 + \sum_{k=1}^{N-1} r'_k I_k + \sum_{k=1}^{N-1} q'_k \Gamma_k + p' \Gamma_N \right] \pi \\
&\quad + \frac{1}{2} \pi' \left( I'_0 R I_0 + \sum_{k=1}^{N-1} \Gamma'_k Q \Gamma_k + \Gamma'_k M I_k + I'_k M' \Gamma_k + I'_k R I_k + \Gamma'_N P \Gamma_N \right) \pi \\
&= \gamma + q' \pi + \frac{1}{2} \pi' Q \pi
\end{aligned} \tag{B.37}$$

By inspection, the parameters in the condensed expression for this quadratic program are

$$\gamma \leftarrow \frac{1}{2} x'_0 \left( \sum_{k=0}^{N-1} (A^k)' Q A^k + (A^N)' P A^N \right) x_0 + \left( \sum_{k=0}^{N-1} (A^k)' q_k + (A^N)' p \right)' x_0 \tag{B.38a}$$

$$\begin{aligned}
q \leftarrow & \left( \sum_{k=0}^{N-1} I'_k M' A^k + \sum_{k=1}^{N-1} \Gamma'_k Q A^k + \Gamma'_N P A^N \right) x_0 + \sum_{k=0}^{N-1} I'_k r_k \\
& + \sum_{k=1}^{N-1} \Gamma'_k q_k + \Gamma'_N p
\end{aligned} \tag{B.38b}$$

$$Q \leftarrow \sum_{k=0}^{N-1} I'_k R I_k + \sum_{k=1}^{N-1} \Gamma'_k Q \Gamma_k + \Gamma'_k M I_k + I'_k M' \Gamma_k + \Gamma'_N P \Gamma_N \tag{B.38c}$$



□

**Remark B.2.14**

The quadratic-term weighting matrix  $Q$  depends on the constant data matrices only. Hence, it can be computed off-line. The linear-term weighting matrix  $q$  depends on the constant data matrices as well as  $x_0$ ,  $\{q_k\}_{k=0}^{N-1}$ ,  $\{r_k\}_{k=0}^{N-1}$ , and  $p$ . Due to the dependence of the latter data,  $q$  must be computed online.

**Remark B.2.15**

By inspection of equation (B.27b) the weighting matrix  $Q_{x_0}$  on  $x_0$  in  $q$  is

$$Q_{x_0} = \sum_{k=0}^{N-1} I'_k M' A^k + \sum_{k=1}^{N-1} \Gamma'_k Q A^k + \Gamma'_N P A^N \quad (\text{B.39})$$

This matrix may be computed off-line.

**Remark B.2.16**

The second term in  $q$  is effectively computed by

$$\sum_{k=0}^{N-1} I'_k r_k = \begin{pmatrix} r_0 \\ r_1 \\ \vdots \\ r_{N-1} \end{pmatrix} \quad (\text{B.40})$$

**Remark B.2.17**

As all non-zero information about  $\Gamma_k$  is contained in  $\Gamma_N$ , only  $\Gamma_N$  need to be computed. This matrix may be computed off-line and is used in all the computations

$$\Gamma_N = \begin{pmatrix} A^{N-1}B & A^{N-2}B & \dots & AB & B \end{pmatrix} \quad (\text{B.41})$$

**Remark B.2.18**

The explicit structure of the matrix  $Q_{x_0}$  is

$$\begin{aligned} Q_{x_0} &= \begin{pmatrix} M' \\ M'A \\ M'A^2 \\ \vdots \\ M'A^{N-2} \\ M'A^{N-1} \end{pmatrix} + \begin{pmatrix} B' \left( \sum_{j=0}^{N-2} (A^j)' Q A^j \right) A \\ B' \left( \sum_{j=0}^{N-3} (A^j)' Q A^j \right) A^2 \\ B' \left( \sum_{j=0}^{N-4} (A^j)' Q A^j \right) A^3 \\ \vdots \\ B' \left( \sum_{j=0}^0 (A^j)' Q A^j \right) A^{N-1} \\ 0 \end{pmatrix} + \begin{pmatrix} B'(A^{N-1})' P A^{N-1} A \\ B'(A^{N-2})' P A^{N-2} A^2 \\ B'(A^{N-3})' P A^{N-3} A^3 \\ \vdots \\ B'A' P A A^{N-1} \\ B' P A^N \end{pmatrix} \\ &= \begin{pmatrix} M' + B'\Psi_{N-1}A \\ M'A + B'\Psi_{N-2}A^2 \\ M'A^2 + B'\Psi_{N-3}A^3 \\ \vdots \\ M'A^{N-2} + B'\Psi_1 A^{N-1} \\ M'A^{N-1} + B'\Psi_0 A^N \end{pmatrix} \end{aligned} \quad (\text{B.42})$$

in which

$$\Psi_0 = P \quad (\text{B.43a})$$

$$\Psi_1 = Q + A'PA \quad (\text{B.43b})$$

$$\Psi_2 = \sum_{j=0}^{2-1} (A^j)'QA^j + (A^2)'PA^2 \quad (\text{B.43c})$$

$$\vdots \quad (\text{B.43d})$$

$$\Psi_{N-1} = \sum_{j=0}^{N-2} (A^j)'QA^j + (A^{N-1})'PA^{N-1} \quad (\text{B.43e})$$

**Remark B.2.19**

The Hessian matrix in the quadratic program may be split into two terms

$$Q = Q^I + Q^{II} \quad (\text{B.44})$$

in which the first term is

$$\begin{aligned} Q^I &= \sum_{k=1}^{N-1} \Gamma'_k Q \Gamma_k + \Gamma'_N P \Gamma_N \\ &= \begin{pmatrix} B'\Psi_{N-1}B & B'A'\Psi_{N-2}B & B'(A^2)'\Psi_{N-3}B & \dots & B'(A^{N-1})'\Psi_0B \\ B'\Psi_{N-2}AB & B'\Psi_{N-2}B & B'A'\Psi_{N-3}B & \dots & B'(A^{N-2})'\Psi_0B \\ B'\Psi_{N-3}A^2B & B'\Psi_{N-3}AB & B'\Psi_{N-3}B & \dots & B'(A^{N-3})'\Psi_0B \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ B'\Psi_0A^{N-1}B & B'\Psi_0A^{N-2}B & B'\Psi_0A^{N-3}B & \dots & B'\Psi_0B \end{pmatrix} \end{aligned} \quad (\text{B.45})$$

and the second term is

$$\begin{aligned} Q^{II} &= \sum_{k=1}^{N-1} \Gamma'_k M I_k + I'_k M' \Gamma_k + \sum_{k=0}^{N-1} I'_k R I_k \\ &= \begin{pmatrix} R & B'M & B'AM & \dots & B'(A^{N-2})'M \\ M'B & R & B'M & \dots & B'(A^{N-3})'M \\ M'AB & M'B & R & \dots & B'(A^{N-4})'M \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ M'A^{N-2}B & M'A^{N-3}B & M'A^{N-4}B & \dots & R \end{pmatrix} \end{aligned} \quad (\text{B.46})$$

The transformation of a model predictive control quadratic program to a standard quadratic program is summarized in the next proposition.

**Proposition B.2.20 (MPC QP as a standard QP)**

The model predictive control quadratic program (B.3) is identical to solving a stan-

dard quadratic program (B.1) with the data matrices defined as

$$q \leftarrow Q_{x0}x_0 + \sum_{k=0}^{N-1} I'_k r_k + \sum_{k=1}^{N-1} \Gamma'_k q_k + \Gamma'_N p \quad (\text{B.47})$$

$$Q_{x0} \leftarrow \sum_{k=0}^{N-1} I'_k M' A^k + \sum_{k=1}^{N-1} \Gamma'_k Q A^k + \Gamma'_N P A^N \quad (\text{B.48})$$

$$Q \leftarrow \sum_{k=0}^{N-1} I'_k R I_k + \sum_{k=1}^{N-1} \Gamma'_k Q \Gamma_k + \Gamma'_k M I_k + I'_k M' \Gamma_k + \Gamma'_N P \Gamma_N \quad (\text{B.49})$$

$$A \leftarrow \begin{pmatrix} H & 0 & 0 & \dots & 0 \\ GB & H & 0 & \dots & 0 \\ GAB & GB & H & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ GA^{N-2}B & GA^{N-3}B & GA^{N-4}B & \dots & H \\ \tilde{G}A^{N-1}B & \tilde{G}A^{N-2}B & \tilde{G}A^{N-3}B & \dots & \tilde{G}B \end{pmatrix} \quad (\text{B.50})$$

$$l \leftarrow \begin{pmatrix} b_l \\ b_l \\ b_l \\ \vdots \\ b_l \\ \tilde{b}_l \end{pmatrix} - \begin{pmatrix} G \\ GA \\ GA^2 \\ \vdots \\ GA^{N-1} \\ \tilde{G}A^N \end{pmatrix} x_0 \quad u \leftarrow \begin{pmatrix} b_u \\ b_u \\ b_u \\ \vdots \\ b_u \\ \tilde{b}_u \end{pmatrix} - \begin{pmatrix} G \\ GA \\ GA^2 \\ \vdots \\ GA^{N-1} \\ \tilde{G}A^N \end{pmatrix} x_0 \quad (\text{B.51})$$

$$\pi_{min} = \begin{pmatrix} u_{min} \\ u_{min} \\ \vdots \\ u_{min} \end{pmatrix} \quad \pi_{max} = \begin{pmatrix} u_{max} \\ u_{max} \\ \vdots \\ u_{max} \end{pmatrix} \quad (\text{B.52})$$

*Proof.* This follows directly from the preceding discussion.  $\square$

**Remark B.2.21**

For notational convenience let  $F$  be the matrix defined by

$$F = \begin{pmatrix} G \\ GA \\ GA^2 \\ \vdots \\ GA^{N-1} \\ \tilde{G}A^N \end{pmatrix} \quad b_{l0} = \begin{pmatrix} b_l \\ b_l \\ b_l \\ \vdots \\ b_l \\ \tilde{b}_l \end{pmatrix} \quad b_{u0} = \begin{pmatrix} b_u \\ b_u \\ b_u \\ \vdots \\ b_u \\ \tilde{b}_u \end{pmatrix} \quad (\text{B.53})$$

Then the lower and upper limits,  $l$  and  $u$ , may be computed as

$$l \leftarrow b_{l0} - Fx_0 \quad u \leftarrow b_{u0} - Fx_0 \quad (\text{B.54})$$

The matrices  $Q$ ,  $Q_{x0}$ ,  $A$ ,  $F$ , and  $\Gamma_N$  may be computed off-line, while the remaining vectors are computed and constructed on-line due to the way data arrives.

## B.3 MPC with Nonzero Exogenous Input

In situations in which an exogenous input sequence  $\{d_k\}_{k=0}^{N-1}$  is known, the model predictive controller formulation must be extended. Exogenous inputs arises for instance when a model is used for computing disturbances or some forecast is used to predict the exogenous input. If only  $d_0$  is measured and used as exogenous input, this extension corresponds to adding a feed-forward mechanism in the model predictive controller. When the entire sequence  $\{d_k\}_{k=0}^{N-1}$  is applied in computation of the control variable,  $u_0^*$ , this type of control is referred to as anticipatory model predictive control. The quadratic program for the linear time invariant model predictive controller with exogenous inputs is stated in the definition below.

**Definition B.3.1 (LTI MPC with Exogenous Inputs Quadratic Program)**

The quadratic program defining the model predictive controller based on a linear time invariant model with exogenous inputs is

$$\min_{\{x_{k+1}, u_k\}_{k=0}^{N-1}} \phi = \frac{1}{2} \sum_{k=0}^{N-1} \{x'_k Q x_k + 2x'_k M u_k + u'_k R u_k + 2q'_k x_k + 2r'_k u_k\} \quad (\text{B.55a})$$

$$+ \frac{1}{2} x'_N P x_N + p' x_N \quad (\text{B.55b})$$

$$s.t. \quad x_{k+1} = A x_k + B u_k + E d_k \quad k = 0, 1, \dots, N-1 \quad (\text{B.55c})$$

$$u_{min} \leq u_k \leq u_{max} \quad k = 0, 1, \dots, N-1 \quad (\text{B.55d})$$

$$b_l \leq G x_k + H u_k \leq b_u \quad k = 0, 1, \dots, N-1 \quad (\text{B.55e})$$

$$\tilde{b}_l \leq \tilde{G} x_N \leq \tilde{b}_u \quad (\text{B.55f})$$

in which the data is  $x_0, \{q_k\}_{k=0}^{N-1}, \{r_k\}_{k=0}^{N-1}, \{d_k\}_{k=0}^{N-1}, p, Q, M, R, P, A, B, E, G, H, b_l, b_u, \tilde{G}, \tilde{b}_l, \tilde{b}_u, u_{min}$ , and  $u_{max}$ .

The solution is both the control sequence  $\{u_k^*\}_{k=0}^{N-1}$  and the state sequence  $\{x_k^*\}_{k=1}^N$ . The corresponding optimal value of the objective function is denoted  $\phi^*$ .

**Remark B.3.2**

The horizon,  $N$ , used for the exogenous input sequence  $\{d_k\}_{k=0}^{N-1}$  is in the formulation equivalent with the control horizon. Normally the control horizon is chosen longer than the horizon for the exogenous input sequence. We have chosen to make them equal to keep the notation simple. If a shorter horizon for the exogenous inputs is chosen, the remaining exogenous inputs may be set to zero; or more effectively the appropriate rows and columns of certain matrices in the subsequent derivation may be simply removed.

For convenience in the subsequent deduction of the standard quadratic program equivalent with (B.55) the sequence of input vectors  $\{d_k\}_{k=0}^{N-1}$  is stacked in a vector.

**Definition B.3.3 (Exogenous Input Vector)**

The exogenous input vector,  $d$ , is the sequence of exogenous input vectors  $\{d_k\}_{k=0}^{N-1}$

stacked according to

$$d = \begin{pmatrix} d_0 \\ d_1 \\ d_2 \\ \vdots \\ d_{N-2} \\ d_{N-1} \end{pmatrix} \quad (\text{B.56})$$

**Definition B.3.4 (Exogenous-Input-to-State Transition Matrix)**

Define the exogenous-input-to-state transition matrix,  $\Gamma_d$ , as

$$\Gamma^d = \begin{pmatrix} \Gamma_0^d \\ \Gamma_1^d \\ \Gamma_2^d \\ \vdots \\ \Gamma_{N-1}^d \\ \Gamma_N^d \end{pmatrix} = \begin{pmatrix} E & 0 & 0 & \dots & 0 & 0 \\ AE & E & 0 & \dots & 0 & 0 \\ A^2E & AE & E & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ A^{N-2}E & A^{N-3}E & A^{N-4}E & \dots & E & 0 \\ A^{N-1}E & A^{N-2}E & A^{N-3}E & \dots & AE & E \end{pmatrix} \quad (\text{B.57})$$

**Remark B.3.5**

The  $k$ th block-row of the exogenous-input-to-state matrix is

$$\Gamma_k^d = (A^{k-1}E \quad A^{k-2}E \quad \dots \quad AE \quad E \quad 0 \quad \dots \quad 0) \quad (\text{B.58})$$

This matrix is important in eliminating the states  $x_k$  from the quadratic program describing the model predictive controller with exogenous inputs.

In the actual computation of the exogenous-input-to-state transition matrix it is sufficient to compute the  $N$ th block-row

$$\Gamma_N^d = (A^{N-1}E \quad A^{N-2}E \quad A^{N-3}E \quad \dots \quad AE \quad E) \quad (\text{B.59})$$

The remaining non-zero blocks in the exogenous-input-to-state transition matrix are computed by appropriate block-copy operations using  $\Gamma_N^d$ .

**Lemma B.3.6 (Expression for the State Dynamics)**

The state dynamics  $x_{k+1} = Ax_k + Bu_k + Ed_k$   $k = 0, 1, \dots, N-1$  implies that the states of the system may be expressed in terms of the initial state  $x_0$ , the control sequence  $\{u_k\}_{k=0}^{N-1}$  and the exogenous input sequence  $\{d_k\}_{k=0}^{N-1}$  according to

$$x_k = A^k x_0 + \sum_{j=0}^{k-1} A^{k-1-j} B u_j + \sum_{j=0}^{k-1} A^{k-1-j} E d_j \quad k = 1, 2, \dots, N \quad (\text{B.60})$$

In using  $\pi$  and  $d$  instead of  $\{u_k\}_{k=0}^{N-1}$  and  $\{d_k\}_{k=0}^{N-1}$ , respectively, the states may be expressed as

$$x_k = A^k x_0 + \Gamma_k \pi + \Gamma_k^d d \quad k = 1, 2, \dots, N \quad (\text{B.61})$$

*Proof.* The proof is similar to the proof of lemma B.2.7.  $\square$

Equation (B.61) is important because it relates the states  $x_k$  to the initial state,  $x_0$ , the control variables,  $\pi$ , and the exogenous inputs,  $d$ , modulo the system dynamics,  $x_{k+1} = Ax_k + Bu_k + Ed_k$ . Hence, it may be used in expressing the general constraints and the objective functions in terms of the known initial state,  $x_0$ , the known exogenous inputs,  $d$ , and the control variables,  $\pi$ . The control variables are the decision variables. As (B.61) by construction satisfies the system dynamics, this constraint can be simply removed from the quadratic program, when the states,  $x_k$ , are removed by the substitution with equation (B.61). Overall this procedure leads to a problem with reduced dimension, both in terms of decision variables and constraints. The first because  $x_k$  is eliminated and the latter because the system dynamics constraints are inherently satisfied by the substitution and therefore removed from the quadratic program.

The resulting expressions obtained when (B.61) is used to eliminate the states,  $x_k$ , in the general constraints and the objective functions are stated in the next two lemmas. The proofs explain the algebra involved in obtaining these expressions.

**Lemma B.3.7 (General Constraints)**

The general constraints in the model predictive controller quadratic program

$$b_l \leq Gx_k + Hu_k \leq b_u \quad k = 0, 1, \dots, N-1 \quad (\text{B.62a})$$

$$\tilde{b}_l \leq \tilde{G}x_N \leq \tilde{b}_u \quad (\text{B.62b})$$

modulo the system dynamics  $x_{k+1} = Ax_k + Bu_k + Ed_k$  may be expressed in terms of  $x_0$ ,  $\pi$ , and  $d$  by

$$b_l - Gx_0 \leq HI_0\pi \leq b_u - Gx_0 \quad (\text{B.63a})$$

$$b_l - GA^k x_0 - G\Gamma_k^d d \leq (G\Gamma_k + HI_k)\pi \leq b_u - GA^k x_0 - \Gamma_k^d d \quad k = 1, 2, \dots, N-1 \quad (\text{B.63b})$$

$$\tilde{b}_l - \tilde{G}A^N x_0 - \tilde{G}\Gamma_N^d d \leq \tilde{G}\Gamma_N\pi \leq \tilde{b}_u - \tilde{G}A^N x_0 - \tilde{G}\Gamma_N^d d \quad (\text{B.63c})$$

*Proof.* The states  $\{x_k\}_{k=1}^N$  and the controls  $\{u_k\}_{k=0}^{N-1}$  are eliminated from the general constraints using  $x_k = A^k x_0 + \Gamma_k \pi + \Gamma_k^d d$  and  $u_k = I_k \pi$ . Subsequent simple rearrangement of terms lead to the desired result.  $\square$

**Lemma B.3.8 (Objective Function)**

The objective function

$$\phi = \frac{1}{2} \left( \sum_{k=0}^{N-1} x_k' Q x_k + 2x_k' M u_k + u_k' R u_k + 2q_k' x_k + 2r_k' u_k \right) + \frac{1}{2} x_N' P x_N + p' x_N \quad (\text{B.64})$$

modulo the system dynamics  $x_{k+1} = Ax_k + Bu_k + Ed_k$  can be expressed as a quadratic function in  $\pi$

$$\phi = \rho + q' \pi + \frac{1}{2} \pi' Q \pi \quad (\text{B.65})$$

in which the coefficients are

$$\begin{aligned} \rho \leftarrow & \frac{1}{2}x'_0 Q x_0 + q'_0 x_0 \\ & + \frac{1}{2} \sum_{k=1}^{N-1} (A^k x_0 + \Gamma_k^d d)' Q (A^k x_0 + \Gamma_k^d d) + 2q'_k (A^k x_0 + \Gamma_k^d d) \\ & + \frac{1}{2} (A^N x_0 + \Gamma_N^d d)' P (A^N x_0 + \Gamma_N^d d) + p' (A^N x_0 + \Gamma_N^d d) \end{aligned} \quad (\text{B.66a})$$

$$\begin{aligned} q \leftarrow & \left( \sum_{k=0}^{N-1} I'_k M' A^k + \sum_{k=1}^{N-1} \Gamma'_k Q A^k + \Gamma'_N P A^N \right) x_0 \\ & + \left( \sum_{k=1}^{N-1} (\Gamma'_k Q + I'_k M') \Gamma_k^d + \Gamma'_N P \Gamma_N^d \right) d \\ & + \sum_{k=0}^{N-1} I'_k r_k + \sum_{k=1}^{N-1} \Gamma'_k q_k + \Gamma'_N p \end{aligned} \quad (\text{B.66b})$$

$$Q \leftarrow \sum_{k=0}^{N-1} I'_k R I_k + \sum_{k=1}^{N-1} \Gamma'_k Q \Gamma_k + \Gamma'_k M I_k + I'_k M' \Gamma_k + \Gamma'_N P \Gamma_N \quad (\text{B.66c})$$

*Proof.* The objective function is

$$\phi = \frac{1}{2} \sum_{k=0}^{N-1} x'_k Q x_k + 2x'_k M u_k + u'_k R u_k + 2q'_k x_k + 2r'_k u_k + \frac{1}{2} x'_N P x_N + p' x_N \quad (\text{B.67})$$

Elimination of  $x_k$  and  $u_k$  in this function is based on the relations

$$u_k = I_k \pi \quad k = 0, 1, \dots, N-1 \quad (\text{B.68})$$

$$x_k = A^k x_0 + \Gamma_k \pi + \Gamma_k^d d \quad k = 1, 2, \dots, N \quad (\text{B.69})$$

For  $k > 0$  the quadratic term  $x'_k Q x_k$  may be expressed as

$$\begin{aligned} x'_k Q x_k &= (A^k x_0 + \Gamma_k \pi + \Gamma_k^d d)' Q (A^k x_0 + \Gamma_k \pi + \Gamma_k^d d) \\ &= (A^k x_0 + \Gamma_k^d d)' Q (A^k x_0 + \Gamma_k^d d) + 2(A^k x_0 + \Gamma_k^d d)' Q \Gamma_k \pi + \pi' \Gamma'_k Q \Gamma_k \pi \end{aligned} \quad (\text{B.70})$$

The mixed term in  $x_k$  and  $u_k$  ( $k > 0$ ) is

$$\begin{aligned} x'_k M u_k &= (A^k x_0 + \Gamma_k \pi + \Gamma_k^d d)' M I_k \pi \\ &= (A^k x_0 + \Gamma_k^d d)' M I_k \pi + \pi' \Gamma'_k M I_k \pi \\ &= x'_0 (A^k)' M I_k \pi + d' (\Gamma_k^d)' M I_k \pi + \pi' \Gamma'_k M I_k \pi \end{aligned} \quad (\text{B.71})$$

The quadratic term in  $u_k$  for  $k \geq 0$  is

$$u'_k R u_k = \pi' I'_k R I_k \pi \quad (\text{B.72})$$

For  $k > 0$  the linear term in the states  $q'_k x_k$  becomes

$$\begin{aligned}
 q'_k x_k &= q'_k (A^k x_0 + \Gamma_k \pi + \Gamma_k^d d) \\
 &= q'_k (A^k x_0 + \Gamma_k^d d) + q'_k \Gamma'_k \pi \\
 &= q'_k A^k x_0 + q'_k \Gamma_k^d d + q'_k \Gamma_k \pi
 \end{aligned} \tag{B.73}$$

The linear term  $r'_k u_k$  in the controls is

$$r'_k u_k = r'_k I_k \pi \tag{B.74}$$

The cost-to-go quadratic term  $x'_N P x_N$  is

$$\begin{aligned}
 x'_N P x_N &= (A^N x_0 + \Gamma_N \pi + \Gamma_N^d d)' P (A^N x_0 + \Gamma_N \pi + \Gamma_N^d d) \\
 &= (A^N x_0 + \Gamma_N^d d)' P (A^N x_0 + \Gamma_N^d d) \\
 &\quad + 2(A^N x_0 + \Gamma_N^d d)' P \Gamma_N \pi + \pi' \Gamma'_N P \Gamma_N \pi
 \end{aligned} \tag{B.75}$$

The cost-to-go linear term  $p' x_N$  is

$$\begin{aligned}
 p' x_N &= p' (A^N x_0 + \Gamma_N \pi + \Gamma_N^d d) \\
 &= p' (A^N x_0 + \Gamma_N^d d) + p' \Gamma_N \pi \\
 &= p' A^N x_0 + p' \Gamma_N^d d + p' \Gamma_N \pi
 \end{aligned} \tag{B.76}$$



Substitution of the above expressions in the objective function (B.67) and subsequent rearrangement of terms yields

$$\begin{aligned}
\phi &= \frac{1}{2} \sum_{k=0}^{N-1} \{x'_k Q x_k + 2x'_k M u_k + u'_k R u_k + 2q'_k x_k + 2r'_k u_k\} + \frac{1}{2} x'_N P x_N + p' x_N \\
&= \frac{1}{2} x'_0 Q x_0 + x'_0 M I_0 \pi + \frac{1}{2} \pi' I'_0 R I_0 \pi + q'_0 x_0 + r'_0 I_0 \pi \\
&\quad + \frac{1}{2} \sum_{k=1}^{N-1} \underbrace{(A^k x_0 + \Gamma_k^d d)' Q (A^k x_0 + \Gamma_k^d d) + 2(A^k x_0 + \Gamma_k^d d)' Q \Gamma_k \pi + \pi' \Gamma'_k Q \Gamma_k \pi}_{=x'_k Q x_k} \\
&\quad + \frac{1}{2} \sum_{k=1}^{N-1} \underbrace{2(A^k x_0 + \Gamma_k^d d)' M I_k \pi + \pi' (\Gamma'_k M I_k + I'_k M' \Gamma_k) \pi}_{=2x'_k M u_k} + \underbrace{\pi' I'_k R I_k \pi}_{=u'_k R u_k} \\
&\quad + \sum_{k=1}^{N-1} \underbrace{q'_k A^k x_0 + q_k \Gamma_k^d d + q'_k \Gamma_k \pi}_{=q'_k x_k} + \underbrace{r'_k I_k \pi}_{=r'_k u_k} \\
&\quad + \frac{1}{2} \underbrace{(A^N x_0 + \Gamma_N^d d)' P (A^N x_0 + \Gamma_N^d d) + (A^N x_0 + \Gamma_N^d d)' P \Gamma_N \pi + \frac{1}{2} \pi' \Gamma'_N P \Gamma_N \pi}_{=x'_N P x_N} \\
&\quad + \underbrace{p' (A^N x_0 + \Gamma_N^d d) + p' \Gamma_N \pi}_{=p' x_N} \\
&= \left\{ \frac{1}{2} x'_0 Q x_0 + q'_0 x_0 + \frac{1}{2} \sum_{k=1}^{N-1} (A^k x_0 + \Gamma_k^d d)' Q (A^k x_0 + \Gamma_k^d d) + 2q'_k (A^k x_0 + \Gamma_k^d d) \right. \\
&\quad \left. + \frac{1}{2} (A^N x_0 + \Gamma_N^d d)' P (A^N x_0 + \Gamma_N^d d) + p' (A^N x_0 + \Gamma_N^d d) \right\} \\
&\quad + \left\{ x'_0 \left( M I_0 + \sum_{k=1}^{N-1} (A^k)' M I_k + \sum_{k=1}^{N-1} (A^k)' Q \Gamma_k + (A^N)' P \Gamma_N \right) \right. \\
&\quad \left. + d' \left( \sum_{k=1}^{N-1} (\Gamma_k^d)' (M I_k + Q \Gamma_k) + (\Gamma_N^d)' P \Gamma_N \right) \right. \\
&\quad \left. + r'_0 I_0 + \sum_{k=1}^{N-1} r'_k I_k + \sum_{k=1}^{N-1} q'_k \Gamma_k + p' \Gamma_N \right\} \pi \\
&\quad + \frac{1}{2} \pi' \left\{ I'_0 R I_0 + \sum_{k=1}^{N-1} I'_k R I_k + \sum_{k=1}^{N-1} \Gamma'_k M I_k + I'_k M' \Gamma_k + \Gamma'_k Q \Gamma_k + \Gamma'_N P \Gamma_N \right\} \pi \\
&= \rho + q' \pi + \frac{1}{2} \pi' Q \pi
\end{aligned} \tag{B.77}$$

By inspection, the zeroth order term is

$$\begin{aligned} \rho \leftarrow & \frac{1}{2} x_0' Q x_0 + q_0' x_0 \\ & + \frac{1}{2} \sum_{k=1}^{N-1} (A^k x_0 + \Gamma_k^d d)' Q (A^k x_0 + \Gamma_k^d d) + 2q_k' (A^k x_0 + \Gamma_k^d d) \\ & + \frac{1}{2} (A^N x_0 + \Gamma_N^d d)' P (A^N x_0 + \Gamma_N^d d) + p' (A^N x_0 + \Gamma_N^d d) \end{aligned} \quad (\text{B.78})$$

The coefficient of the linear term is obtained by transposition of the linear coefficient in the above expression. Hence

$$\begin{aligned} q \leftarrow & \left( \sum_{k=0}^{N-1} I_k' M' A^k + \sum_{k=1}^{N-1} \Gamma_k' Q A^k + \Gamma_N' P A^N \right) x_0 \\ & + \left( \sum_{k=1}^{N-1} (\Gamma_k' Q + I_k' M') \Gamma_k^d + \Gamma_N' P \Gamma_N^d \right) d \\ & + \sum_{k=0}^{N-1} I_k' r_k + \sum_{k=1}^{N-1} \Gamma_k' q_k + \Gamma_N' p \end{aligned} \quad (\text{B.79})$$

The weight matrix for the quadratic term is also obtained by direct inspection in the last expression of  $\phi$

$$Q \leftarrow \sum_{k=0}^{N-1} I_k' R I_k + \left( \sum_{k=1}^{N-1} \Gamma_k' Q \Gamma_k + \Gamma_k' M I_k + I_k' M' \Gamma_k \right) + \Gamma_N' P \Gamma_N \quad (\text{B.80})$$

□

The computation of the data matrices in a dense quadratic program that corresponds to a model predictive controller with exogenous inputs is stated in the following proposition. This is a key result.

**Proposition B.3.9 (Standard QP matrices, LTI MPC with Exo. Inputs)**

The data in the standard quadratic program (B.1) for the model predictive controller with exogenous inputs (B.55) are

$$Q = \sum_{k=0}^{N-1} I_k' R I_k + \sum_{k=1}^{N-1} \Gamma_k' Q \Gamma_k + \Gamma_k' M I_k + I_k' M' \Gamma_k + \Gamma_N' P \Gamma_N \quad (\text{B.81a})$$

$$\begin{aligned} q = & \left( \sum_{k=0}^{N-1} I_k' M' A^k + \sum_{k=1}^{N-1} \Gamma_k' Q A^k + \Gamma_N' P A^N \right) x_0 \\ & + \left( \sum_{k=1}^{N-1} (\Gamma_k' Q + I_k' M') \Gamma_k^d + \Gamma_N' P \Gamma_N^d \right) d \\ & + \sum_{k=0}^{N-1} I_k' r_k + \sum_{k=1}^{N-1} \Gamma_k' q_k + \Gamma_N' p \end{aligned} \quad (\text{B.81b})$$

$$A = \begin{pmatrix} H & 0 & 0 & \dots & 0 & 0 \\ GB & H & 0 & \dots & 0 & 0 \\ GAB & GB & H & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ GA^{N-2}B & GA^{N-3}B & GA^{N-4}B & \dots & GB & H \\ \tilde{G}A^{N-1}B & \tilde{G}A^{N-2}B & \tilde{G}A^{N-3}B & \dots & \tilde{G}AB & \tilde{G}B \end{pmatrix} \quad (\text{B.81c})$$

$$l = \begin{pmatrix} b_l \\ b_l \\ b_l \\ \vdots \\ b_l \\ \tilde{b}_l \end{pmatrix} - \begin{pmatrix} G \\ GA \\ GA^2 \\ \vdots \\ GA^{N-1} \\ \tilde{G}A^N \end{pmatrix} x_0 \quad (\text{B.81d})$$

$$- \begin{pmatrix} 0 & 0 & 0 & \dots & 0 & 0 \\ GE & 0 & 0 & \dots & 0 & 0 \\ GAE & GE & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ GA^{N-2}E & GA^{N-3}E & GA^{N-4}E & \dots & GE & 0 \\ \tilde{G}A^{N-1}E & \tilde{G}A^{N-2}E & \tilde{G}A^{N-3}E & \dots & \tilde{G}AE & \tilde{G}E \end{pmatrix} d$$

$$u = \begin{pmatrix} b_u \\ b_u \\ b_u \\ \vdots \\ b_u \\ \tilde{b}_u \end{pmatrix} - \begin{pmatrix} G \\ GA \\ GA^2 \\ \vdots \\ GA^{N-1} \\ \tilde{G}A^N \end{pmatrix} x_0 \quad (\text{B.81e})$$

$$- \begin{pmatrix} 0 & 0 & 0 & \dots & 0 & 0 \\ GE & 0 & 0 & \dots & 0 & 0 \\ GAE & GE & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ GA^{N-2}E & GA^{N-3}E & GA^{N-4}E & \dots & GE & 0 \\ \tilde{G}A^{N-1}E & \tilde{G}A^{N-2}E & \tilde{G}A^{N-3}E & \dots & \tilde{G}AE & \tilde{G}E \end{pmatrix} d$$

$$\pi_{min} = \begin{pmatrix} u_{min} \\ u_{min} \\ \vdots \\ u_{min} \end{pmatrix} \quad \pi_{max} = \begin{pmatrix} u_{max} \\ u_{max} \\ \vdots \\ u_{max} \end{pmatrix} \quad (\text{B.81f})$$

*Proof.* Follows directly from the preceding results.  $\square$

### Corollary B.3.10

The coefficient vector for the linear term in the objective function of (B.1) for the model predictive controller with exogenous inputs is

$$q = Q_{x_0}x_0 + Q_d d + \sum_{k=0}^{N-1} I'_k r_k + \sum_{k=1}^{N-1} \Gamma'_k q_k + \Gamma'_N p \quad (\text{B.82})$$

in which

$$Q_{x_0} = \sum_{k=0}^{N-1} I'_k M' A^k + \sum_{k=1}^{N-1} \Gamma'_k Q A^k + \Gamma'_N P A^N \quad (\text{B.83a})$$

$$Q_d = \sum_{k=1}^{N-1} (\Gamma'_k Q + I'_k M') \Gamma_k^d + \Gamma'_N P \Gamma_N^d \quad (\text{B.83b})$$

*Proof.* Follows by inspection of equation (B.81b).  $\square$

## B.4 MPC with Time Variant Linear Models

Linear time variant models arise as subproblems in solution of nonlinear optimal control problems as well as in the control of batch and fed-batch processes. The linear time variant MPC quadratic program is defined in the following proposition.

### Definition B.4.1 (Linear Time-Variant MPC Quadratic Program)

The quadratic program defining the model predictive controller with a linear time variant model and exogenous inputs is

$$\min_{\{x_{k+1}, u_k\}_{k=0}^{N-1}} \phi = \frac{1}{2} \sum_{k=0}^{N-1} \{x'_k Q_k x_k + 2x'_k M_k u_k + u'_k R_k u_k + 2q'_k x_k + 2r'_k u_k\} \quad (\text{B.84a})$$

$$+ \frac{1}{2} x'_N P x_N + p' x_N \quad (\text{B.84b})$$

$$s.t. \quad x_{k+1} = A_k x_k + B_k u_k + b_k \quad k = 0, 1, \dots, N-1 \quad (\text{B.84c})$$

$$u_{min} \leq u_k \leq u_{max} \quad k = 0, 1, \dots, N-1 \quad (\text{B.84d})$$

$$b_{l,k} \leq G_k x_k + H_k u_k \leq b_{u,k} \quad k = 0, 1, \dots, N-1 \quad (\text{B.84e})$$

$$\tilde{b}_l \leq \tilde{G} x_N \leq \tilde{b}_u \quad (\text{B.84f})$$

in which the data is  $x_0$ ,  $\{q_k\}_{k=0}^{N-1}$ ,  $\{r_k\}_{k=0}^{N-1}$ ,  $p$ ,  $\{Q_k, M_k, R_k\}_{k=0}^{N-1}$ ,  $P$ ,  $\{A_k, B_k\}_{k=0}^{N-1}$ ,  $\{G_k, H_k, b_{l,k}, b_{u,k}\}_{k=0}^{N-1}$ ,  $\tilde{G}$ ,  $\tilde{b}_l$ ,  $\tilde{b}_u$ ,  $u_{min}$ , and  $u_{max}$ .

The solution is both the control sequence  $\{u_k^*\}_{k=0}^{N-1}$  and the state sequence  $\{x_k\}_{k=1}^N$ . The corresponding optimal value of the objective function is denoted  $\phi^*$ .

To facilitate expressing the quadratic program for the time variant linear model predictive controller with exogenous inputs as a standard quadratic program, we must introduce some basic concepts from linear systems theory (c.f. Rugh, 1996).

### Definition B.4.2 (State Transition Matrix)

The state transition matrix for the linear time variant system  $x_{k+1} = A_k x_k + B_k u_k + b_k$  is defined in terms of the matrix sequence  $\{A_k\}_{k=0}^\infty$  as

$$\Phi(k, j) = \begin{cases} A_{k-1} A_{k-2} \dots A_j & k > j \\ I & k = j \end{cases} \quad (\text{B.85})$$

**Remark B.4.3**

For notational convenience we adopt the notation  $\Phi_{k,j} = \Phi(k, j)$ .

**Remark B.4.4**

The state space transition matrix stems from the time variant system  $x_{k+1} = A_k x_k$  as  $x_k = A_{k-1} A_{k-2} \dots A_j x_j = \Phi_{k,j} x_j$ . This notational effective way of representing the transition from  $x_j$  to  $x_k$  is the primary reason for introducing the state transition matrix.

For the subsequent derivations relating the states  $x_k$  to the initial state  $x_0$ , the input sequence  $\{u_k\}_{k=0}^{N-1}$ , and the exogenous input sequence  $\{b_k\}_{k=0}^{N-1}$ , the following lemma is needed

**Lemma B.4.5**

$$\Phi(k+1, j) = A_k \Phi(k, j) \quad k \geq j \quad (\text{B.86})$$

*Proof.*

$$\begin{aligned} k > j : \quad A_k \Phi(k, j) &= A_k (A_{k-1} A_{k-2} \dots A_j) \\ &= A_k A_{k-1} A_{k-2} \dots A_j = \Phi(k+1, j) \end{aligned} \quad (\text{B.87})$$

$$k = j : \quad A_k \Phi(k, j) = A_k I = A_k = \Phi(k+1, k) = \Phi(k+1, j) \quad (\text{B.88})$$

□

**Definition B.4.6 (Linear Time Variant Input-to-State Transition Matrix)**

The input-to-state transition matrix for a linear time variant system  $x_{k+1} = A_k x_k + B_k u_k + b_k$  is

$$\begin{aligned} \Gamma &= \begin{pmatrix} \Gamma_1 \\ \Gamma_2 \\ \Gamma_3 \\ \vdots \\ \Gamma_{N-1} \\ \Gamma_N \end{pmatrix} \\ &= \begin{pmatrix} \Phi_{1,1} B_0 & 0 & 0 & \dots & 0 & 0 \\ \Phi_{2,1} B_0 & \Phi_{2,2} B_1 & 0 & \dots & 0 & 0 \\ \Phi_{3,1} B_0 & \Phi_{3,2} B_1 & \Phi_{3,3} B_2 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ \Phi_{N-1,1} B_0 & \Phi_{N-1,2} B_1 & \Phi_{N-1,3} B_2 & \dots & \Phi_{N-1,N-1} B_{N-2} & 0 \\ \Phi_{N,1} B_0 & \Phi_{N,2} B_1 & \Phi_{N,3} B_2 & \dots & \Phi_{N,N-1} B_{N-2} & \Phi_{N,N} B_{N-1} \end{pmatrix} \end{aligned} \quad (\text{B.89})$$

**Remark B.4.7**

The  $k$ th block-row of the input-to-state matrix is

$$\Gamma_k = (\Phi_{k,1} B_0 \quad \Phi_{k,2} B_1 \quad \Phi_{k,3} B_2 \quad \dots \quad \Phi_{k,k-1} B_{k-2} \quad \Phi_{k,k} B_{k-1} \quad 0 \quad \dots \quad 0) \quad (\text{B.90})$$

This matrix is important in eliminating the states  $x_k$  from the quadratic program describing the model predictive controller for the time variant model with exogenous inputs.

In the actual computation of the input-to-state transition matrix each block column may be computed recursively. However all columns need to be computed due to the time variant property of the system.

**Definition B.4.8 (Exogenous Input Vector)**

The exogenous input-vector is

$$b = \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ \vdots \\ b_{N-2} \\ b_{N-1} \end{pmatrix} \quad (\text{B.91})$$

**Definition B.4.9 (Exogenous-Input-to-State Transition Matrix)**

The exogenous-input-to-state matrix  $\Gamma_b$  of the system  $x_{k+1} = A_k x_k + B_k u_k + b_k$  is

$$\Gamma^b = \begin{pmatrix} \Gamma_1^b \\ \Gamma_2^b \\ \Gamma_3^b \\ \vdots \\ \Gamma_{N-1}^b \\ \Gamma_N^b \end{pmatrix} = \begin{pmatrix} \Phi_{1,1} & 0 & 0 & \dots & 0 & 0 \\ \Phi_{2,1} & \Phi_{2,2} & 0 & \dots & 0 & 0 \\ \Phi_{3,1} & \Phi_{3,2} & \Phi_{3,3} & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ \Phi_{N-1,1} & \Phi_{N-1,2} & \Phi_{N-1,3} & \dots & \Phi_{N-1,N-1} & 0 \\ \Phi_{N,1} & \Phi_{N,2} & \Phi_{N,3} & \dots & \Phi_{N,N-1} & \Phi_{N,N} \end{pmatrix} \quad (\text{B.92})$$

**Remark B.4.10**

The  $k$ th block row of the exogenous-input-to-state transition matrix is

$$\Gamma_k^b = (\Phi_{k,1} \quad \Phi_{k,2} \quad \Phi_{k,3} \quad \dots \quad \Phi_{k,k-1} \quad \Phi_{k,k} \quad 0 \quad \dots \quad 0) \quad (\text{B.93})$$

This matrix is important in eliminating the states  $x_k$  from the quadratic program describing the model predictive controller with linear time variant model and exogenous inputs.

In the actual computation of the exogenous-input-to-state transition matrix each block-column may be computed recursively. But unlike the time invariant case, all columns need to be computed due to the time variant property of the model.

**Lemma B.4.11 (Expression for the State Dynamics)**

The state dynamics  $x_{k+1} = A_k x_k + B_k u_k + b_k$  for  $k = 0, 1, \dots, N-1$  implies that the states  $x_k$  ( $k = 1, 2, \dots, N$ ) of the system may be expressed in terms of the initial state  $x_0$ , the control sequence  $\{u_k\}_{k=0}^{N-1}$ , and the exogenous input sequence  $\{b_k\}_{k=0}^{N-1}$  according to

$$x_k = \Phi_{k,0} x_0 + \sum_{j=0}^{k-1} \Phi_{k,j+1} B_j u_j + \sum_{j=0}^{k-1} \Phi_{k,j+1} b_j \quad (\text{B.94})$$

In using  $\pi$  and  $b$  instead of  $\{u_k\}_{k=0}^{N-1}$  and  $\{b_k\}_{k=0}^{N-1}$ , respectively, the states may be expressed as

$$x_k = \Phi_{k,0} x_0 + \Gamma_k \pi + \Gamma_k^b b \quad (\text{B.95})$$

*Proof.* The first part of the proof concerns establishing that the state dynamics  $x_{k+1} = A_k x_k + B_k u_k + b_k$  implies  $x_k = \Phi_{k,0} x_0 + \sum_{j=0}^{k-1} \Phi_{k,j+1} B_j u_j + \sum_{j=0}^{k-1} \Phi_{k,j+1} b_j$ . The proof is by recursion. Note that the statement

$$P(k) : \quad x_k = \Phi_{k,0} x_0 + \sum_{j=0}^{k-1} \Phi_{k,j+1} B_j u_j + \sum_{j=0}^{k-1} \Phi_{k,j+1} b_j \quad (\text{B.96})$$

is true for  $k = 1$ , i.e.

$$P(k=1) : \quad x_1 = A_0 x_0 + B_0 u_0 + b_0 \quad (\text{B.97})$$

$$= \Phi_{1,0} x_0 + \Phi_{1,1} B_0 u_0 + \Phi_{1,1} b_0 \quad (\text{B.98})$$

$$= \Phi_{1,0} x_0 + \sum_{j=0}^{1-1} \Phi_{1,j+1} B_j u_j + \sum_{j=0}^{1-1} \Phi_{1,j+1} b_j \quad (\text{B.99})$$

Next assume that the statement  $P(k)$  is true. We want to show that this implies that  $P(k+1)$  is also true. For convenience let  $l = k+1$ . The system dynamics  $x_{k+1} = A_k x_k + B_k u_k + b_k$  and  $P(k)$  implies

$$\begin{aligned} x_l &= x_{k+1} = A_k x_k + B_k u_k + b_k \\ &= A_k \left( \Phi_{k,0} x_0 + \sum_{j=0}^{k-1} \Phi_{k,j+1} B_j u_j + \sum_{j=0}^{k-1} \Phi_{k,j+1} b_j \right) + \Phi_{k+1,k+1} B_k u_k + \Phi_{k+1,k+1} b_k \\ &= \Phi_{k+1,0} x_0 + \sum_{j=0}^{k-1} \Phi_{k+1,j+1} B_j u_j + \Phi_{k+1,k+1} B_k u_k + \sum_{j=0}^{k-1} \Phi_{k+1,j+1} b_j + \Phi_{k+1,k+1} b_k \\ &= \Phi_{l,0} x_0 + \sum_{j=0}^{l-1} \Phi_{l,j+1} B_j u_j + \sum_{j=0}^{l-1} \Phi_{l,j+1} b_j \end{aligned} \quad (\text{B.100})$$

As  $P(1)$  is true and  $P(k) \implies P(k+1)$ ,  $P(k)$  is true for all  $k \in \{1, 2, \dots\}$ . If we adopt the convention  $\sum_j^k(\cdot) = 0$  whenever  $k < j$  then  $P(0)$  is also true.

The next part of the proof concerns establishing that  $x_k = \Phi_{k,0} x_0 + \Gamma_k \pi + \Gamma_k^b b$  is identical with the expression  $x_k = \Phi_{k,0} x_0 + \sum_{j=0}^{k-1} \Phi_{k,j+1} B_j u_j + \sum_{j=0}^{k-1} \Phi_{k,j+1} b_j$ . The proof is accomplished by reformulation of the last two sums. The first sum

is

$$\begin{aligned}
& \sum_{j=0}^{k-1} \Phi_{k,j+1} B_j u_j = \\
& (\Phi_{k,1} B_0 \quad \Phi_{k,2} B_1 \quad \Phi_{k,3} B_2 \quad \dots \quad \Phi_{k,k-1} B_{k-2} \quad \Phi_{k,k} B_{k-1} \quad 0 \quad \dots \quad 0) \begin{pmatrix} u_0 \\ u_1 \\ u_2 \\ \dots \\ u_{k-2} \\ u_{k-1} \\ u_k \\ \vdots \\ u_{N-1} \end{pmatrix} \\
& = \Gamma_k \pi
\end{aligned} \tag{B.101}$$

and the second sum is

$$\begin{aligned}
& \sum_{j=0}^{k-1} \Phi_{k,j+1} b_j = (\Phi_{k,1} \quad \Phi_{k,2} \quad \Phi_{k,3} \quad \dots \quad \Phi_{k,k-1} \quad \Phi_{k,k} \quad 0 \quad \dots \quad 0) \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ \vdots \\ b_{k-2} \\ b_{k-1} \\ b_k \\ \vdots \\ b_{N-1} \end{pmatrix} \\
& = \Gamma_k^b b
\end{aligned} \tag{B.102}$$

Hence, we have the expression

$$\begin{aligned}
x_k &= \Phi_{k,0} x_0 + \sum_{j=0}^{k-1} \Phi_{k,j+1} B_j u_j + \sum_{j=0}^{k-1} \Phi_{k,j+1} b_j \\
&= \Phi_{k,0} x_0 + \Gamma_k \pi + \Gamma_k^b b
\end{aligned} \tag{B.103}$$

□

**Lemma B.4.12 (General Constraints)**

The general constraints in the time variant linear model predictive controller

$$b_{l,k} \leq G_k x_k + H_k u_k \leq b_{u,k} \tag{B.104a}$$

$$\tilde{b}_l \leq \tilde{G} x_k \leq \tilde{b}_u \tag{B.104b}$$



modulo the system dynamics  $x_{k+1} = A_k x_k + B_k u_k + b_k$  may be expressed in terms of  $x_0$ ,  $\pi$ , and  $b$  by the equations

$$b_{l,0} - G_0 x_0 \leq H_0 I_0 \pi \leq b_{u,0} - G_0 x_0 \quad (\text{B.105a})$$

$$b_{l,k} - G_k \Phi_{k,0} x_0 - G_k \Gamma_k^b b \leq (G_k \Gamma_k + H_k I_k) \pi \leq b_{u,k} - G_k \Phi_{k,0} x_0 - G_k \Gamma_k^b b \quad (\text{B.105b})$$

$$\tilde{b}_l - \tilde{G} \Phi_{N,0} x_0 - \tilde{G} \Gamma_N^b b \leq \tilde{G} \Gamma_N \pi \leq \tilde{b}_u - \tilde{G} \Phi_{N,0} x_0 - \tilde{G} \Gamma_N^b b \quad (\text{B.105c})$$

*Proof.* The results are obtained by substitution of the expressions  $x_k = \Phi_{k,0} x_0 + \Gamma_k \pi + \Gamma_k^b b$  and  $u_k = I_k \pi$  for  $x_k$  and  $u_k$  in the general constraints. Subsequent rearrangement of terms leads to the desired result.  $\square$

#### Lemma B.4.13 (Objective Function)

The objective function of the time variant linear model predictive controller with exogenous inputs

$$\phi = \frac{1}{2} \left( \sum_{k=0}^{N-1} x'_k Q_k x_k + 2x'_k M_k u_k + u'_k R_k u_k + 2q'_k x_k + 2r'_k u_k \right) + \frac{1}{2} x'_N P x_N + p' x_N \quad (\text{B.106})$$

modulo the system dynamics  $x_{k+1} = A_k x_k + B_k u_k + b_k$  may be expressed as

$$\phi = \rho + q' \pi + \frac{1}{2} \pi' Q \pi \quad (\text{B.107})$$

in which the coefficients are

$$\begin{aligned} \rho \leftarrow & \frac{1}{2} x'_0 Q_0 x_0 + q'_0 x_0 \\ & + \frac{1}{2} \left( \sum_{k=1}^{N-1} (\Phi_{k,0} x_0 + \Gamma_k^b b)' Q_k (\Phi_{k,0} x_0 + \Gamma_k^b b) + 2q'_k (\Phi_{k,0} x_0 + \Gamma_k^b b) \right) \\ & + \frac{1}{2} (\Phi_{N,0} x_0 + \Gamma_N^b b)' P (\Phi_{N,0} x_0 + \Gamma_N^b b) + p' (\Phi_{N,0} x_0 + \Gamma_N^b b) \end{aligned} \quad (\text{B.108})$$

$$\begin{aligned} q \leftarrow & \left( \sum_{k=0}^{N-1} I'_k M'_k \Phi_{k,0} + \sum_{k=1}^{N-1} \Gamma'_k Q_k \Phi_{k,0} + \Gamma'_N P \Phi_{N,0} \right) x_0 \\ & + \left( \sum_{k=1}^{N-1} \Gamma'_k Q_k \Gamma_k^b + I'_k M'_k \Gamma_k^b + \Gamma'_N P \Gamma_N^b \right) b \\ & + \sum_{k=0}^{N-1} I'_k r_k + \sum_{k=1}^{N-1} \Gamma'_k q_k + \Gamma'_N p \end{aligned} \quad (\text{B.109})$$

$$Q \leftarrow \sum_{k=0}^{N-1} I'_k R_k I_k + \sum_{k=1}^{N-1} \Gamma'_k M_k I_k + I'_k M'_k \Gamma_k + \Gamma'_k Q_k \Gamma_k + \Gamma'_N P \Gamma_N \quad (\text{B.110})$$

*Proof.* The objective function of the model predictive controller based on a time variant linear model with exogenous inputs is

$$\begin{aligned} \phi = & \frac{1}{2} \left( \sum_{k=0}^{N-1} x'_k Q_k x_k + 2x'_k M_k u_k + u'_k R_k u_k + 2q'_k x_k + 2r'_k u_k \right) \\ & + \frac{1}{2} x'_N P x_N + p' x_N \end{aligned} \quad (\text{B.111})$$

The dynamic equation  $x_{k+1} = A_k x_k + B_k u_k + b_k$  means that the states may be expressed in terms of  $x_0$ ,  $\pi$ , and  $b$ :

$$x_k = \Phi_{k,0} x_0 + \Gamma_k \pi + \Gamma_k^b b \quad k = 1, 2, \dots, N \quad (\text{B.112})$$

The inputs  $u_k$  are related to  $\pi$  by

$$u_k = I_k \pi \quad k = 0, 1, \dots, N-1 \quad (\text{B.113})$$

Next we express each term in the objective function in terms of  $x_0$ ,  $\pi$ , and  $b$ . The quadratic term  $x_k' Q_k x_k$  for  $k = 1, 2, \dots, N$  becomes

$$\begin{aligned} x_k' Q_k x_k &= (\Phi_{k,0} x_0 + \Gamma_k \pi + \Gamma_k^b b)' Q_k (\Phi_{k,0} x_0 + \Gamma_k \pi + \Gamma_k^b b) \\ &= (\Phi_{k,0} x_0 + \Gamma_k^b b)' Q_k (\Phi_{k,0} x_0 + \Gamma_k^b b) \\ &\quad + 2(\Phi_{k,0} x_0 + \Gamma_k^b b)' Q_k \Gamma_k \pi + \pi' \Gamma_k' Q_k \Gamma_k \pi \end{aligned} \quad (\text{B.114})$$

The mixed term in  $x_k$  and  $u_k$ ,  $x_k' M_k u_k$ , may be expressed as

$$\begin{aligned} x_k' M_k u_k &= (\Phi_{k,0} x_0 + \Gamma_k \pi + \Gamma_k^b b)' M_k I_k \pi \\ &= (\Phi_{k,0} x_0 + \Gamma_k^b b)' M_k I_k \pi + \pi' \Gamma_k' M_k I_k \pi \\ &= (\Phi_{k,0} x_0 + \Gamma_k^b b)' M_k I_k \pi + \frac{1}{2} \pi' (\Gamma_k' M_k I_k + I_k' M_k' \Gamma_k) \pi \end{aligned} \quad (\text{B.115})$$

The quadratic term  $u_k' R_k u_k$  is in terms of  $\pi$

$$u_k' R_k u_k = \pi' I_k' R_k I_k \pi \quad (\text{B.116})$$

The linear term  $q_k' x_k$  becomes

$$\begin{aligned} q_k' x_k &= q_k' (\Phi_{k,0} x_0 + \Gamma_k \pi + \Gamma_k^b b) \\ &= q_k' (\Phi_{k,0} x_0 + \Gamma_k^b b) + q_k' \Gamma_k \pi \end{aligned} \quad (\text{B.117})$$

The linear term  $r_k' u_k$  is

$$r_k' u_k = r_k' I_k \pi \quad (\text{B.118})$$

The quadratic cost-to-go term  $x_N' P x_N$  becomes

$$\begin{aligned} x_N' P x_N &= (\Phi_{N,0} x_0 + \Gamma_N \pi + \Gamma_N^b b)' P (\Phi_{N,0} x_0 + \Gamma_N \pi + \Gamma_N^b b) \\ &= (\Phi_{N,0} x_0 + \Gamma_N^b b)' P (\Phi_{N,0} x_0 + \Gamma_N^b b) \\ &\quad + 2(\Phi_{N,0} x_0 + \Gamma_N^b b)' P \Gamma_N \pi + \pi' \Gamma_N' P \Gamma_N \pi \end{aligned} \quad (\text{B.119})$$

The linear cost-to-go term  $p' x_N$  is

$$\begin{aligned} p' x_N &= p' (\Phi_{N,0} x_0 + \Gamma_N \pi + \Gamma_N^b b) \\ &= p' (\Phi_{N,0} x_0 + \Gamma_N^b b) + p' \Gamma_N \pi \end{aligned} \quad (\text{B.120})$$

Upon substitution of the above expressions in the objective function it becomes

$$\begin{aligned}
\phi &= \frac{1}{2}x'_0Q_0x_0 + x'_0M_0u_0 + \frac{1}{2}u'_0R_0u_0 + q'_0x_0 + r'_0u_0 \\
&\quad + \frac{1}{2}\sum_{k=1}^{N-1}x'_kQ_kx_k + 2x'_kM_ku_k + u'_kR_ku_k + 2q'_kx_k + 2r_ku_k \\
&\quad + \frac{1}{2}x'_N Px_N + p'x_N \\
&= \frac{1}{2}x'_0Q_0x_0 + x'_0M_0I_0\pi + \frac{1}{2}\pi'I'_0R_0I_0\pi + q'_0x_0 + r'_0I_0\pi \\
&\quad + \frac{1}{2}\sum_{k=1}^{N-1}\underbrace{(\Phi_{k,0}x_0 + \Gamma_k^b b)'Q_k(\Phi_{k,0}x_0 + \Gamma_k^b b) + 2(\Phi_{k,0}x_0 + \Gamma_k^b b)'Q_k\Gamma_k\pi + \pi'\Gamma_k'Q_k\Gamma_k\pi}_{=x'_kQ_kx_k} \\
&\quad + \sum_{k=1}^{N-1}\underbrace{(\Phi_{k,0}x_0 + \Gamma_k^b b)'M_kI_k\pi + \frac{1}{2}\pi'(\Gamma_k'M_kI_k + I_k'M_k'\Gamma_k)\pi}_{=x'_kM_ku_k} \\
&\quad + \frac{1}{2}\sum_{k=1}^{N-1}\underbrace{\pi'\Gamma_k'R_k\Gamma_k\pi}_{=u'_kR_ku_k} + \sum_{k=1}^{N-1}\underbrace{q'_k(\Phi_{k,0}x_0 + \Gamma_k^b b)' + q'_k\Gamma_k\pi}_{=q'_kx_k} + \sum_{k=1}^{N-1}\underbrace{r'_kI_k\pi}_{=r'_ku_k} \\
&\quad + \frac{1}{2}\underbrace{\left((\Phi_{N,0}x_0 + \Gamma_N^b b)'P(\Phi_{N,0}x_0 + \Gamma_N^b b) + 2(\Phi_{N,0}x_0 + \Gamma_N^b b)'P\Gamma_N\pi + \pi'\Gamma_N'P\Gamma_N\pi\right)}_{=x'_N Px_N} \\
&\quad + \underbrace{p'(\Phi_{N,0}x_0 + \Gamma_N^b b) + p'\Gamma_N\pi}_{=p'x_N}
\end{aligned} \tag{B.121}$$

which by collection and rearrangement yields

$$\begin{aligned}
\phi &= \left\{ \frac{1}{2}x'_0Q_0x_0 + q'_0x_0 \right. \\
&\quad + \frac{1}{2}\sum_{k=1}^{N-1}(\Phi_{k,0}x_0 + \Gamma_k^b b)'Q_k(\Phi_{k,0}x_0 + \Gamma_k^b b) + 2q'_k(\Phi_{k,0}x_0 + \Gamma_k^b b) \\
&\quad \left. + \frac{1}{2}(\Phi_{N,0}x_0 + \Gamma_N^b b)'P(\Phi_{N,0}x_0 + \Gamma_N^b b) + p'(\Phi_{N,0}x_0 + \Gamma_N^b b) \right\} \\
&\quad + \left\{ \left( \sum_{k=0}^{N-1}I_k'M_k'\Phi_{k,0} + \sum_{k=1}^{N-1}\Gamma_k'Q_k\Phi_{k,0} + \Gamma_N'P\Phi_{N,0} \right) x_0 \right. \\
&\quad + \left( \sum_{k=1}^{N-1}\Gamma_k'Q_k\Gamma_k^b + I_k'M_k'\Gamma_k^b + \Gamma_N'P\Gamma_N^b \right) b + \sum_{k=0}^{N-1}I_k'r_k + \sum_{k=1}^{N-1}\Gamma_k'q_k + \Gamma_N'p \left. \right\}' \pi \\
&\quad + \frac{1}{2}\pi' \left\{ \sum_{k=0}^{N-1}I_k'R_kI_k + \sum_{k=1}^{N-1}\Gamma_k'M_kI_k + I_k'M_k'\Gamma_k + \Gamma_k'Q_k\Gamma_k + \Gamma_N'P\Gamma_N \right\} \pi \\
&= \rho + q'\pi + \frac{1}{2}\pi'Q\pi
\end{aligned} \tag{B.122}$$

By inspection in this expression, the coefficients are

$$\begin{aligned} \rho \leftarrow & \frac{1}{2}x'_0Q_0x_0 + q'_0x_0 \\ & + \frac{1}{2}\left(\sum_{k=1}^{N-1}(\Phi_{k,0}x_0 + \Gamma_k^b b)'Q_k(\Phi_{k,0}x_0 + \Gamma_k^b b) + 2q'_k(\Phi_{k,0}x_0 + \Gamma_k^b b)\right) \\ & + \frac{1}{2}(\Phi_{N,0}x_0 + \Gamma_N^b b)'P(\Phi_{N,0}x_0 + \Gamma_N^b b) + p'(\Phi_{N,0}x_0 + \Gamma_N^b b) \end{aligned} \quad (\text{B.123})$$

$$\begin{aligned} q \leftarrow & \left(\sum_{k=0}^{N-1}I'_kM'_k\Phi_{k,0} + \sum_{k=1}^{N-1}\Gamma'_kQ_k\Phi_{k,0} + \Gamma'_NP\Phi_{N,0}\right)x_0 \\ & + \left(\sum_{k=1}^{N-1}\Gamma'_kQ_k\Gamma_k^b + I'_kM'_k\Gamma_k^b + \Gamma'_NP\Gamma_N^b\right)b \\ & + \sum_{k=0}^{N-1}I'_kr_k + \sum_{k=1}^{N-1}\Gamma'_kq_k + \Gamma'_Np \end{aligned} \quad (\text{B.124})$$

$$Q \leftarrow \sum_{k=0}^{N-1}I'_kR_kI_k + \sum_{k=1}^{N-1}\Gamma'_kM_kI_k + I'_kM'_k\Gamma_k + \Gamma'_kQ_k\Gamma_k + \Gamma'_NP\Gamma_N \quad (\text{B.125})$$

□

The coefficient vector for the linear term may be arranged differently. This is the content of the next corollary.

**Corollary B.4.14**

The coefficient to the linear term in the objective function is

$$q = Q_{x_0}x_0 + Q_b b + \sum_{k=0}^{N-1}I'_kr_k + \sum_{k=1}^{N-1}\Gamma'_kq_k + \Gamma'_Np \quad (\text{B.126})$$

in which

$$Q_{x_0} = \sum_{k=0}^{N-1}I'_kM'_k\Phi_{k,0} + \sum_{k=1}^{N-1}\Gamma'_kQ_k\Phi_{k,0} + \Gamma'_NP\Phi_{N,0} \quad (\text{B.127})$$

$$Q_b = \sum_{k=1}^{N-1}\Gamma'_kQ_k\Gamma_k^b + I'_kM'_k\Gamma_k^b + \Gamma'_NP\Gamma_N^b \quad (\text{B.128})$$

*Proof.* The result follows directly by inspection in the expression

$$\begin{aligned}
q = & \left( \sum_{k=0}^{N-1} I'_k M'_k \Phi_{k,0} + \sum_{k=1}^{N-1} \Gamma'_k Q_k \Phi_{k,0} + \Gamma'_N P \Phi_{N,0} \right) x_0 \\
& + \left( \sum_{k=1}^{N-1} \Gamma'_k Q_k \Gamma_k^b + I'_k M'_k \Gamma_k^b + \Gamma'_N P \Gamma_N^b \right) b \\
& + \sum_{k=0}^{N-1} I'_k r_k + \sum_{k=1}^{N-1} \Gamma'_k q_k + \Gamma'_N p
\end{aligned} \tag{B.129}$$

□

The term independent of  $\pi$  in the objective function is addressed in the next corollary. It concerns the computation of  $\rho$  given  $x_0$  and  $b$ .

**Corollary B.4.15**

The constant term,  $\rho$ , in the objective function may be expressed as

$$\rho = \frac{1}{2} x'_0 W x_0 + w' x_0 + x'_0 S b + \frac{1}{2} b' V b + v' b \tag{B.130}$$

in which the matrices are

$$W = \sum_{k=0}^{N-1} \Phi'_{k,0} Q_k \Phi_{k,0} + \Phi'_{N,0} P \Phi_{N,0} \tag{B.131}$$

$$S = \sum_{k=1}^{N-1} \Phi'_{k,0} Q_k \Gamma_k^b + \Phi'_{N,0} P \Gamma_N^b \tag{B.132}$$

$$V = \sum_{k=1}^{N-1} (\Gamma_k^b)' Q_k \Gamma_k^b + (\Gamma_N^b)' P \Gamma_N^b \tag{B.133}$$

$$\tag{B.134}$$

and the vectors are

$$w = \sum_{k=0}^{N-1} \Phi'_{k,0} q_k + \Phi'_{N,0} p \tag{B.135}$$

$$v = \sum_{k=1}^{N-1} (\Gamma_k^b)' q_k + (\Gamma_N^b)' p \tag{B.136}$$

*Proof.* The zeroth order term is

$$\begin{aligned}
\rho = & \frac{1}{2} x'_0 Q_0 x_0 + q'_0 x_0 \\
& + \frac{1}{2} \left\{ \sum_{k=1}^{N-1} (\Phi_{k,0} x_0 + \Gamma_k^b b)' Q_k (\Phi_{k,0} x_0 + \Gamma_k^b b) + 2 q'_k (\Phi_{k,0} x_0 + \Gamma_k^b b) \right\} \\
& + \frac{1}{2} (\Phi_{N,0} x_0 + \Gamma_N^b b)' P (\Phi_{N,0} x_0 + \Gamma_N^b b) + p' (\Phi_{N,0} x_0 + \Gamma_N^b b)
\end{aligned} \tag{B.137}$$

Each individual term in this expression is expanded to subsequently collect quadratic and linear terms in  $x_0$ , quadratic and linear terms in  $b$  as well as cross terms in  $x_0$  and  $b$ .

$$\begin{aligned}
 (\Phi_{k,0}x_0 + \Gamma_k^b b)' Q_k (\Phi_{k,0}x_0 + \Gamma_k^b b) = \\
 x_0' \Phi_{k,0}' Q_k \Phi_{k,0} x_0 + 2x_0' \Phi_{k,0}' Q_k \Gamma_k^b b + b' (\Gamma_k^b)' Q_k \Gamma_k^b b
 \end{aligned} \tag{B.138}$$

$$q_k'(\Phi_{k,0}x_0 + \Gamma_k^b b) = q_k' \Phi_{k,0} x_0 + q_k' \Gamma_k^b b \tag{B.139}$$

$$\begin{aligned}
 (\Phi_{N,0}x_0 + \Gamma_N^b b)' P (\Phi_{N,0}x_0 + \Gamma_N^b b) = \\
 x_0' \Phi_{N,0}' P \Phi_{N,0} x_0 + 2x_0' \Phi_{N,0}' P \Gamma_N^b b + b' (\Gamma_N^b)' P \Gamma_N^b b
 \end{aligned} \tag{B.140}$$

$$p'(\Phi_{N,0}x_0 + \Gamma_N^b b) = p' \Phi_{N,0} x_0 + p' \Gamma_N^b b \tag{B.141}$$

$$\begin{aligned}
\rho &= \frac{1}{2}x'_0Q_0x_0 + q'_0x_0 \\
&\quad + \frac{1}{2}\left\{\sum_{k=1}^{N-1}(\Phi_{k,0}x_0 + \Gamma_k^b b)'Q_k(\Phi_{k,0}x_0 + \Gamma_k^b b) + 2q'_k(\Phi_{k,0}x_0 + \Gamma_k^b b)\right\} \\
&\quad + \frac{1}{2}(\Phi_{N,0}x_0 + \Gamma_N^b b)P(\Phi_{N,0}x_0 + \Gamma_N^b b) + p'(\Phi_{N,0}x_0 + \Gamma_N^b b) \\
&= \frac{1}{2}x'_0Q_0x_0 + q'_0x_0 \\
&\quad + \frac{1}{2}\sum_{k=1}^{N-1}\underbrace{x'_0\Phi'_{k,0}Q_k\Phi_{k,0}x_0 + 2x'_0\Phi'_{k,0}Q_k\Gamma_k^b b + b'(\Gamma_k^b)'Q_k\Gamma_k^b b}_{=(\Phi_{k,0}x_0 + \Gamma_k^b b)'Q_k(\Phi_{k,0}x_0 + \Gamma_k^b b)} \\
&\quad + \sum_{k=1}^{N-1}\underbrace{q'_k\Phi_{k,0}x_0 + q'_k\Gamma_k^b b}_{=q'_k(\Phi_{k,0}x_0 + \Gamma_k^b b)} \\
&\quad + \frac{1}{2}\underbrace{(x'_0\Phi'_{N,0}P\Phi_{N,0}x_0 + 2x'_0\Phi'_{N,0}P\Gamma_N^b b + b'(\Gamma_N^b)'P\Gamma_N^b b)}_{=(\Phi_{N,0}x_0 + \Gamma_N^b b)'P(\Phi_{N,0}x_0 + \Gamma_N^b b)} + \underbrace{p'\Phi_{N,0}x_0 + p'\Gamma_N^b b}_{=p'(\Phi_{N,0}x_0 + \Gamma_N^b b)} \\
&= \frac{1}{2}x'_0\left\{\underbrace{Q_0 + \sum_{k=1}^{N-1}\Phi'_{k,0}Q_k\Phi_{k,0} + \Phi'_{N,0}P\Phi_{N,0}}_{=W}\right\}x_0 \\
&\quad + \underbrace{\left\{q'_0 + \sum_{k=1}^{N-1}q'_k\Phi_{k,0} + p'\Phi_{N,0}\right\}}_{=w'}x_0 \\
&\quad + x'_0\left\{\underbrace{\sum_{k=1}^{N-1}\Phi'_{k,0}Q_k\Gamma_k^b + \Phi'_{N,0}P\Gamma_N^b}_{=S}\right\}b \\
&\quad + \frac{1}{2}b'\left\{\underbrace{\sum_{k=1}^{N-1}(\Gamma_k^b)'Q_k\Gamma_k^b + (\Gamma_N^b)'P\Gamma_N^b}_{=V}\right\}b + \underbrace{\left\{\sum_{k=1}^{N-1}q'_k\Gamma_k^b + p'\Gamma_N^b\right\}}_{=v'}b \\
&= \frac{1}{2}x'_0Wx_0 + w'x_0 + x_0Sb + \frac{1}{2}b'Vb + v'b
\end{aligned}$$

(B.142)

in which the quadratic weight matrices are

$$W = Q_0 + \sum_{k=1}^{N-1} \Phi'_{k,0} Q_k \Phi_{k,0} + \Phi'_{N,0} P \Phi_{N,0} \quad (\text{B.143})$$

$$S = \sum_{k=1}^{N-1} \Phi'_{k,0} Q_k \Gamma_k^b + \Phi'_{N,0} P \Gamma_N^b \quad (\text{B.144})$$

$$V = \sum_{k=1}^{N-1} (\Gamma_k^b)' Q_k \Gamma_k^b + (\Gamma_N^b)' P \Gamma_N^b \quad (\text{B.145})$$

and

$$w = \sum_{k=0}^{N-1} \Phi'_{k,0} q_k + \Phi'_{N,0} p \quad (\text{B.146})$$

$$v = \sum_{k=1}^{N-1} (\Gamma_k^b)' q_k + (\Gamma_N^b)' p \quad (\text{B.147})$$

□

The computation of the data matrices in the dense quadratic program for the linear time variant model predictive control quadratic program is summarized in the following proposition.

**Proposition B.4.16 (Standard QP Matrices for LTV MPC QP)**

The data in the standard quadratic program (B.1) for the linear time variant MPC quadratic program (B.84) are

$$Q \leftarrow \sum_{k=0}^{N-1} I'_k R_k I_k + \sum_{k=1}^{N-1} \Gamma'_k M_k I_k + I'_k M'_k \Gamma_k + \Gamma'_k Q_k \Gamma_k + \Gamma'_N P \Gamma_N \quad (\text{B.148})$$

$$q \leftarrow Q_{x_0} x_0 + Q_b b + \sum_{k=0}^{N-1} I'_k r_k + \sum_{k=1}^{N-1} \Gamma'_k q_k + \Gamma'_N p \quad (\text{B.149a})$$

$$Q_{x_0} \leftarrow \sum_{k=0}^{N-1} I'_k M'_k \Phi_{k,0} + \sum_{k=1}^{N-1} \Gamma'_k Q_k \Phi_{k,0} + \Gamma'_N P \Phi_{N,0} \quad (\text{B.149b})$$

$$Q_b \leftarrow \sum_{k=1}^{N-1} \Gamma'_k Q_k \Gamma_k^b + I'_k M'_k \Gamma_k^b + \Gamma'_N P \Gamma_N^b \quad (\text{B.149c})$$

$$A \leftarrow \begin{pmatrix} H_0 & 0 & \dots & 0 & 0 \\ G_1 \Phi_{1,1} B_0 & H_1 & \dots & 0 & 0 \\ G_2 \Phi_{2,1} B_0 & G_2 \Phi_{2,2} B_1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ G_{N-2} \Phi_{N-2,1} B_0 & G_{N-2} \Phi_{N-2,2} B_1 & \dots & H_{N-2} & 0 \\ G_{N-1} \Phi_{N-1,1} B_0 & G_{N-1} \Phi_{N-1,2} B_1 & \dots & G_{N-1} \Phi_{N-1,N-1} B_{N-2} & H_{N-1} \\ G_{N,1} B_0 & G_{N,2} B_1 & \dots & G_{N,N-1} B_{N-2} & G_{N,N} B_{N-1} \end{pmatrix} \quad (\text{B.150})$$

$$l \leftarrow b_{l0} - F_{x_0} x_0 - F_b b \quad u \leftarrow b_{u0} - F_{x_0} x_0 - F_b b \quad (\text{B.151})$$



$$F_{x_0} \leftarrow \begin{pmatrix} G_0 \\ G_1 \Phi_{1,0} \\ G_2 \Phi_{2,0} \\ \vdots \\ G_{N-2} \Phi_{N-2,0} \\ G_{N-1} \Phi_{N-1,0} \\ \tilde{G} \Phi_{N,0} \end{pmatrix} \quad b_{l0} \leftarrow \begin{pmatrix} b_{l,0} \\ b_{l,1} \\ b_{l,2} \\ \vdots \\ b_{l,N-2} \\ b_{l,N-1} \\ \tilde{b}_l \end{pmatrix} \quad b_{u0} \leftarrow \begin{pmatrix} b_{u,0} \\ b_{u,1} \\ b_{u,2} \\ \vdots \\ b_{u,N-2} \\ b_{u,N-1} \\ \tilde{b}_u \end{pmatrix} \quad (\text{B.152})$$

$$F_b \leftarrow \begin{pmatrix} 0 & 0 & \dots & 0 & 0 \\ G_1 \Phi_{1,1} & 0 & \dots & 0 & 0 \\ G_2 \Phi_{2,1} & G_2 \Phi_{2,2} & \dots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ G_{N-2} \Phi_{N-2,1} & G_{N-2} \Phi_{N-2,2} & \dots & 0 & 0 \\ G_{N-1} \Phi_{N-1,1} & G_{N-1} \Phi_{N-1,2} & \dots & G_{N-1} \Phi_{N-1,N-1} & 0 \\ \tilde{G} \Phi_{N,1} & \tilde{G} \Phi_{N,2} & \dots & \tilde{G} \Phi_{N,N-1} & \tilde{G} \Phi_{N,N} \end{pmatrix} \quad (\text{B.153})$$

$$\pi_{min} \leftarrow \begin{pmatrix} u_{min} \\ u_{min} \\ \vdots \\ u_{min} \end{pmatrix} \quad \pi_{max} \leftarrow \begin{pmatrix} u_{max} \\ u_{max} \\ \vdots \\ u_{max} \end{pmatrix} \quad (\text{B.154})$$

$\Gamma_k, \Gamma_k^b$  and  $\Phi_{k,j}$  are defined by (B.90), (B.93), and (B.85), respectively.

*Proof.* Follows directly from the preceding results.  $\square$

# C

## Maximal Output Admissible Set

In a seminal paper Gilbert and Tan (1991) defined the concept of maximum output admissible sets of linear systems and gave algorithms for its computation. This chapter will repeat those results in a setting that is relevant for linear model predictive control. In particular we will restrict ourselves to polytopic output constraints, i.e. output sets  $\mathbb{Y}$  with linear inequality constraints only  $\mathbb{Y} = \{y \in \mathbb{R}^p \mid y_{min} \leq y \leq y_{max}\}$ . From a computational perspective this simplification is important as it implies that the maximum output admissible set can be found by solving a sequence of linear programs. Furthermore for linear model predictive control this is the form of the constraints. So this assumption does not restrict the application.

### C.1 Motivation for Maximal Output Admissible Sets

The output admissibility concept arises for systems with output constraints. Consider the linear system

$$x_{k+1} = Ax_k \tag{C.1}$$

$$y_k = Cx_k \tag{C.2}$$

with the output constraints

$$y_k \in \mathbb{Y} \tag{C.3}$$

In general not all initial states  $x_0$  will satisfy the output constraints at all times. Even initial states  $x_0$  that satisfy the output constraint at time  $k = 0$  may at some subsequent time fail to satisfy the output constraints. An initial state  $x_0$  that does satisfy the output constraints at present (time  $k=0$ ) and all subsequent times is said to be *output admissible* of the system. Any set of output admissible points is said to be *positively invariant* (c.f. Blanchini, 1999). The set of all output admissible states is called the *maximum output admissible*

*set.* The maximum output admissible set is thus the largest positively invariant set of the system.

The relevance of this problem stems from linear control with constraints on the inputs and/or the outputs. Consider the linear system

$$x_{k+1} = Ax_k + Bu_k \quad (\text{C.4})$$

controlled by a linear controller

$$u_k = Kx_k \quad (\text{C.5})$$

with constraints on the actuator and possible also on the output. The constraints may be modelled by the inclusion

$$Cx_k + Du_k \in \mathbb{Y} \quad (\text{C.6})$$

Actuator constraints arise due to limits on the actuator range, while state constraints arise because of product specifications and limitations on the desired operating regime. For this problem the maximal output admissible set characterizes the initial states  $x_0$  for which these actuator limits and state constraints are observed.

This problem may be cast into the standard form by the assignments  $A \leftarrow A + BK$  and  $C \leftarrow C + DK$ . In linear model predictive control knowledge of the maximal output admissible set enables a finite parametrization of an infinite number of constraints.

## C.2 Definitions and Basic Properties

Let  $\mathbb{I}$  denote the set of non-negative integers, i.e.  $\mathbb{I} = \{0\} \cup \mathbb{N}$ , and let  $\mathbb{B}(x_0, r) = \{x \in \mathbb{R}^n \mid \|x - x_0\| < r\}$  denote the open ball with center in  $x_0$  and radius  $r$ . Let  $\mathbb{S} \subset \mathbb{R}^n$  and  $P \in \mathbb{R}^{m \times n}$  then  $P\mathbb{S} = \{z \in \mathbb{R}^m \mid z = Px, x \in \mathbb{S}\}$ .

The tripple  $(A, C, \mathbb{Y})$  denotes the system

$$x_{k+1} = Ax_k \quad (\text{C.7})$$

$$y_k = Cx_k \quad (\text{C.8})$$

with the output constraints

$$y_k \in \mathbb{Y} \quad k \in \mathbb{I} \quad (\text{C.9})$$

The output constraints considered are assumed to be polyhedral

$$\mathbb{Y} = \{y \in \mathbb{R}^p \mid y_{\min} \leq y \leq y_{\max}\} \quad (\text{C.10})$$

### Definition C.2.1 (State Constraint Set)

The state constraint set associated with  $C$  and  $\mathbb{Y}$  is

$$\mathbb{X}(C, \mathbb{Y}) = \{x \in \mathbb{R}^n \mid Cx \in \mathbb{Y}\} \quad (\text{C.11})$$

**Definition C.2.2 (Output Admissible Point)**

A point  $x_0 \in \mathbb{R}^n$  is an output admissible point of the system

$$x_{k+1} = Ax_k \quad (C.12)$$

$$y_k = Cx_k \quad (C.13)$$

with respect to the output constraints  $\mathbb{Y}$  if

$$y_k \in \mathbb{Y} \quad \forall k \in \mathbb{I} \quad (C.14)$$

**Corollary C.2.3**

A point  $x_0 \in \mathbb{R}^n$  is output admissible of the system  $(A, C, \mathbb{Y})$  if and only if

$$CA^k x_0 \in \mathbb{Y} \quad \forall k \in \mathbb{I} \quad (C.15)$$

*Proof.* The output of the system  $(A, C, \mathbb{Y})$  is by simple algebra expressed in terms of the initial state as  $x_k = A^k x_0$  which gives  $y_k = Cx_k = CA^k x_0 \in \mathbb{Y}$ .  $\square$

**Definition C.2.4 (Maximal Output Admissible Set)**

The maximal output admissible set is the set of all output admissible points

$$\mathbb{O}_\infty(A, C, \mathbb{Y}) = \left\{ x \in \mathbb{R}^n \mid CA^k x \in \mathbb{Y} \quad k \in \mathbb{I} \right\} \quad (C.16)$$

**Definition C.2.5 (A-Invariant Set)**

A set  $\mathbb{S} \subset \mathbb{R}^n$  is  $A$ -invariant if  $x \in \mathbb{S} \Rightarrow Ax \in \mathbb{S}$  (i.e.  $A\mathbb{S} \subset \mathbb{S}$ ).

**Definition C.2.6 (Positively Invariant Set)**

A set  $\mathbb{S} \subset \mathbb{R}^n$  is positively invariant with respect to the system  $x_{k+1} = Ax_k$  if  $x_0 \in \mathbb{S} \Rightarrow x_k \in \mathbb{S} \forall k \in \mathbb{I}$ .

As the next lemma states, positive invariance of a set with respect to a linear discrete-time system is equivalent to  $A$ -invariance. This is a special result applying to linear discrete-time systems only. It does not carry over to linear continuous-time systems and obviously not to nonlinear systems.

**Lemma C.2.7**

A set  $\mathbb{S} \subset \mathbb{R}^n$  is positively invariant with respect to  $x_{k+1} = Ax_k$  if and only if it is  $A$ -invariant.

*Proof.* Assume that  $\mathbb{S}$  is positively invariant. Then for  $x_0 \in \mathbb{S} : x_1 = Ax_0 \in \mathbb{S}$  which means that  $\mathbb{S}$  is  $A$ -invariant. Conversely, assume that  $\mathbb{S}$  is  $A$ -invariant, then  $x_k \in \mathbb{S} \Rightarrow x_{k+1} = Ax_k \in \mathbb{S}$ . By induction  $x_0 \in \mathbb{S} \Rightarrow x_k = A^k x_0 \in \mathbb{S} \forall k \in \mathbb{I}$  which means that  $\mathbb{S}$  is positively invariant.  $\square$

**Lemma C.2.8**

Assume that  $\mathbb{S}$  is positively invariant and that  $\mathbb{S} \subset \mathbb{X}(C, \mathbb{Y})$ . Then  $\mathbb{S} \subset \mathbb{O}(A, C, \mathbb{Y})$

*Proof.*  $\forall x \in \mathbb{X}(C, \mathbb{Y}) : Cx \in \mathbb{Y}$  and  $\forall x \in \mathbb{S} : A^k x \in \mathbb{S} \forall k \in \mathbb{I}$ . Consequently  $\forall x \in \mathbb{S} \subset \mathbb{X}(C, \mathbb{Y}) : CA^k x \in \mathbb{Y} \forall k \in \mathbb{I}$  which implies  $x \in \mathbb{S} \subset \mathbb{X}(C, \mathbb{Y}) \Rightarrow x \in \mathbb{O}_\infty(A, C, \mathbb{Y})$ . Thus  $\mathbb{S} \subset \mathbb{O}(A, C, \mathbb{Y})$ .  $\square$

**Remark C.2.9**

Due to lemma C.2.8 and the fact that  $\mathbb{O}_\infty(A, C, \mathbb{Y})$  is positively invariant by construction, the maximal output admissible set is in a certain sense the largest positive invariant set of the system  $(A, C, \mathbb{Y})$ .

**Definition C.2.10**

$$\mathbb{O}_t(A, C, \mathbb{Y}) = \left\{ x \in \mathbb{R}^n \mid CA^k x \in \mathbb{Y} \quad k \in \{0, 1, \dots, t\} \right\} \quad (\text{C.17})$$

**Definition C.2.11 (Finitely Determined Output Admissible Set)**

The output admissible set  $\mathbb{O}_\infty(A, C, \mathbb{Y})$  is finitely determined if

$$\exists t \in \mathbb{I} : \mathbb{O}_\infty(A, C, \mathbb{Y}) = \mathbb{O}_t(A, C, \mathbb{Y}) \quad (\text{C.18})$$

**Definition C.2.12 (Output Admissibility Index)**

The output admissibility index of a finitely determined output admissible set is the integer

$$t^* = \min \{ t \in \mathbb{I} \mid \mathbb{O}_\infty(A, C, \mathbb{Y}) = \mathbb{O}_t(A, C, \mathbb{Y}) \} \quad (\text{C.19})$$

**Lemma C.2.13**

$$\mathbb{O}_\infty(A, C, \mathbb{Y}) \subset \mathbb{O}_{t_2}(A, C, \mathbb{Y}) \subset \mathbb{O}_{t_1}(A, C, \mathbb{Y}) \subset \mathbb{X}(C, \mathbb{Y}) \quad t_2 \geq t_1 \quad t_1, t_2 \in \mathbb{I} \quad (\text{C.20})$$

*Proof.* The proves of all three subset relations are similar and use the set property  $\mathbb{A} = \mathbb{B} \cap \mathbb{C} \subset \mathbb{B}$ .

$$\begin{aligned} \mathbb{O}_\infty(A, C, \mathbb{Y}) &= \{ x \in \mathbb{R}^n \mid CA^k x \in \mathbb{Y} \quad k \in \mathbb{I} \} \\ &= \{ x \in \mathbb{R}^n \mid CA^k x \in \mathbb{Y} \quad k \in \{0, 1, \dots, t_2\} \} \cap \\ &\quad \{ x \in \mathbb{R}^n \mid CA^k x \in \mathbb{Y} \quad k \in \mathbb{I} \wedge k > t_2 \} \\ &= \mathbb{O}_{t_2}(A, C, \mathbb{Y}) \cap \{ x \in \mathbb{R}^n \mid CA^k x \in \mathbb{Y} \quad k \in \mathbb{I} \wedge k > t_2 \} \\ &\subset \mathbb{O}_{t_2}(A, C, \mathbb{Y}) \end{aligned} \quad (\text{C.21})$$

$$\begin{aligned} \mathbb{O}_{t_2}(A, C, \mathbb{Y}) &= \{ x \in \mathbb{R}^n \mid CA^k x \in \mathbb{Y} \quad k \in \{0, 1, \dots, t_2\} \} \\ &= \{ x \in \mathbb{R}^n \mid CA^k x \in \mathbb{Y} \quad k \in \{0, 1, \dots, t_1\} \} \cap \\ &\quad \{ x \in \mathbb{R}^n \mid CA^k x \in \mathbb{Y} \quad k \in \{t_1 + 1, \dots, t_2\} \} \\ &= \mathbb{O}_{t_1}(A, C, \mathbb{Y}) \cap \{ x \in \mathbb{R}^n \mid CA^k x \in \mathbb{Y} \quad k \in \{t_1 + 1, \dots, t_2\} \} \\ &\subset \mathbb{O}_{t_1}(A, C, \mathbb{Y}) \end{aligned} \quad (\text{C.22})$$

$$\begin{aligned}
\mathbb{O}_{t_1}(A, C, \mathbb{Y}) &= \{x \in \mathbb{R}^n \mid CA^k x \in \mathbb{Y} \ k \in \{0, 1, \dots, t_1\}\} \\
&= \{x \in \mathbb{R}^n \mid CA^k x \in \mathbb{Y} \ k = 0\} \cap \\
&\quad \{x \in \mathbb{R}^n \mid CA^k x \in \mathbb{Y} \ k \in \{1, \dots, t_1\}\} \\
&= \{x \in \mathbb{R}^n \mid Cx \in \mathbb{Y}\} \cap \{x \in \mathbb{R}^n \mid CA^k x \in \mathbb{Y} \ k \in \{1, \dots, t_1\}\} \\
&= \mathbb{X}(C, \mathbb{Y}) \cap \{x \in \mathbb{R}^n \mid CA^k x \in \mathbb{Y} \ k \in \{1, \dots, t_1\}\} \\
&\subset \mathbb{X}(C, \mathbb{Y})
\end{aligned} \tag{C.23}$$

□

**Remark C.2.14**

In early literature (c.f. Bertsekas, 1972; Bertsekas and Rhodes, 1971a,b) addressing polyhedral invariant sets, the property  $\mathbb{O}_\infty(A, C, \mathbb{Y}) = \bigcap_{t=0}^\infty \mathbb{O}_t(A, C, \mathbb{Y})$  was used to derive algorithms for its determination. This property follows immediately from lemma C.2.13. Later literature (c.f. Gutman and Cwikel, 1986; Cwikel and Gutman, 1986; Gutman and Cwikel, 1987; Keerthi and Gilbert, 1987) has also focused on this property.

**Theorem C.2.15 (Finitely Determined Output Admissible Set)**

$\mathbb{O}_\infty(A, C, \mathbb{Y})$  is finitely determined if and only if

$$\exists t \in \mathbb{I} : \mathbb{O}_{t+1}(A, C, \mathbb{Y}) = \mathbb{O}_t(A, C, \mathbb{Y}) \tag{C.24}$$

*Proof.*

Assume that  $\mathbb{O}_\infty(A, C, \mathbb{Y})$  is finitely determined. Then  $\exists t \in \mathbb{I} : \mathbb{O}_t(A, C, \mathbb{Y}) = \mathbb{O}_\infty(A, C, \mathbb{Y})$ . This and lemma C.2.13 gives  $\mathbb{O}_\infty(A, C, \mathbb{Y}) \subset \mathbb{O}_{t+1}(A, C, \mathbb{Y}) \subset \mathbb{O}_t(A, C, \mathbb{Y}) = \mathbb{O}_\infty(A, C, \mathbb{Y})$  which implies  $\exists t \in \mathbb{I} : \mathbb{O}_{t+1}(A, C, \mathbb{Y}) = \mathbb{O}_t(A, C, \mathbb{Y})$ .

Assume  $\exists t \in \mathbb{I} : \mathbb{O}_{t+1}(A, C, \mathbb{Y}) = \mathbb{O}_t(A, C, \mathbb{Y})$ . Then  $\mathbb{O}_t(A, C, \mathbb{Y})$  is  $A$ -invariant, i.e.  $x \in \mathbb{O}_t(A, C, \mathbb{Y}) \Rightarrow Ax \in \mathbb{O}_t(A, C, \mathbb{Y})$ . By lemma C.2.13  $\mathbb{O}_t(A, C, \mathbb{Y}) \subset \mathbb{X}(C, \mathbb{Y})$ . By lemma C.2.7 and lemma C.2.8 these properties imply that  $\mathbb{O}_t(A, C, \mathbb{Y})$  is output admissible, i.e.  $\mathbb{O}_t(A, C, \mathbb{Y}) \subset \mathbb{O}_\infty(A, C, \mathbb{Y})$ . The output admissibility and lemma C.2.13 yield  $\mathbb{O}_\infty(A, C, \mathbb{Y}) \subset \mathbb{O}_t(A, C, \mathbb{Y}) \subset \mathbb{O}_\infty(A, C, \mathbb{Y})$ . Thus  $\exists t \in \mathbb{I} : \mathbb{O}_t(A, C, \mathbb{Y}) = \mathbb{O}_\infty(A, C, \mathbb{Y})$  and the maximal output admissible set is finitely determined. □

Theorem C.2.15 suggests a conceptual method for determination of the maximal output admissible set

1.  $t = 0$
2. If  $\mathbb{O}_{t+1}(A, C, \mathbb{Y}) = \mathbb{O}_t(A, C, \mathbb{Y})$  Stop:  $t^* = t$  and  $\mathbb{O}_\infty(A, C, \mathbb{Y}) = \mathbb{O}_t(A, C, \mathbb{Y})$ .
3.  $t = t + 1$ . Goto 2.

The method has two shortcomings. Firstly, it does not prescribe a method to establish whether the sets  $\mathbb{O}_{t+1}(A, C, \mathbb{Y})$  and  $\mathbb{O}_t(A, C, \mathbb{Y})$  are identical. Secondly, it does not prescribe under what conditions the algorithm will stop in

finite time. It only says that if the maximal output admissible set is finitely determined, the algorithm will find it.

The next theorem addresses the latter concern, and gives sufficient conditions for finite determination of the maximal output admissible set. In order to establish the theorem, the following two lemmas must be apprehended.

**Lemma C.2.16**

If  $(A, C)$  is observable and  $\mathbb{Y}$  is bounded then  $\mathbb{O}_\infty(A, C, \mathbb{Y})$  is bounded.

*Proof.* The observability of  $(A, C)$  implies that the observability matrix

$$\mathcal{O} = \begin{pmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{pmatrix} \in \mathbb{R}^{np \times n}$$

has rank  $n$ . Thus if  $\mathcal{O}x = y$  has a solution, the solution is unique, and given by  $x = \mathcal{O}^+ y$ , where  $\mathcal{O}^+ = (\mathcal{O}^T \mathcal{O})^{-1} \mathcal{O}^T$  is the pseudo-inverse. Applying lemma C.2.13 gives

$$\begin{aligned} \mathbb{O}_\infty(A, C, \mathbb{Y}) &\subset \mathbb{O}_{n-1}(A, C, \mathbb{Y}) \\ &= \{x \in \mathbb{R}^n \mid CA^k x \in \mathbb{Y} \ k \in \{0, 1, \dots, n-1\}\} \\ &= \{x \in \mathbb{R}^n \mid \mathcal{O}x \in \mathbb{Y}^n\} \\ &= \mathcal{O}^+ \mathbb{Y}^n \end{aligned}$$

Since  $\mathbb{Y}$  is bounded so are  $\mathcal{O}^+ \mathbb{Y}^n$  and  $\mathbb{O}_\infty(A, C, \mathbb{Y})$ . □

**Lemma C.2.17**

If  $A$  is stable and  $0 \in \text{int}(\mathbb{Y})$  then  $0 \in \text{int}(\mathbb{O}_\infty(A, C, \mathbb{Y}))$

*Proof.*

The stability of  $A$  implies

$$\exists \gamma_1 > 0 \forall x \in \mathbb{R}^n \forall k \in \mathbb{I} : \quad \|CA^k x\| < \gamma_1 \|x\|$$

As  $0 \in \text{int}(\mathbb{Y})$ :

$$\exists \gamma_2 > 0 : \quad \mathbb{B}(0, \gamma_2) \subset \mathbb{Y}$$

From these implications, the following statement holds

$$\|CA^k x\| < \gamma_1 \|x\| < \gamma_2 \Rightarrow CA^k x \in \mathbb{B}(0, \gamma_2) \subset \mathbb{Y} \quad \forall k \in \mathbb{I}$$

which is equivalent to

$$x \in \mathbb{B}\left(0, \frac{\gamma_2}{\gamma_1}\right) \Rightarrow x \in \mathbb{O}_\infty(A, C, \mathbb{Y})$$

Consequently  $\mathbb{B}(0, \frac{\gamma_2}{\gamma_1}) \subset \mathbb{O}_\infty(A, C, \mathbb{Y})$  and  $0 \in \text{int}(\mathbb{O}_\infty(A, C, \mathbb{Y}))$  □

**Theorem C.2.18 (Sufficient Conditions for Finite Determination of  $\mathbb{O}_\infty$ )**

If

1.  $A$  is asymptotically stable
2.  $(A, C)$  is observable
3.  $\mathbb{Y}$  is bounded
4.  $0 \in \text{int}(\mathbb{Y})$

then  $\mathbb{O}_\infty(A, C, \mathbb{Y})$  is finitely determined.

*Proof.*

We must establish  $\exists k \in \mathbb{I} : \mathbb{O}_{k+1}(A, C, \mathbb{Y}) = \mathbb{O}_k(A, C, \mathbb{Y})$ . Lemma C.2.13 gives  $\mathbb{O}_{k+1}(A, C, \mathbb{Y}) \subset \mathbb{O}_k(A, C, \mathbb{Y})$ . Therefore the remaining part of the proof need to establish  $\mathbb{O}_k(A, C, \mathbb{Y}) \subset \mathbb{O}_{k+1}(A, C, \mathbb{Y})$  for some  $k \in \mathbb{I}$ .

As  $(A, C)$  is observable and  $\mathbb{Y}$  is bounded so is  $\mathbb{O}_\infty(A, C, \mathbb{Y})$  according to lemma C.2.16. This implies

$$\exists r \in (0, \infty) : \quad \mathbb{O}_k(A, C, \mathbb{Y}) \subset \mathbb{B}(0, r) \quad \forall k \in \{k \in \mathbb{I} \mid k \geq n-1\}$$

The asymptotic stability of  $A$  implies

$$\forall \gamma > 0 \exists N \in \mathbb{I} : \quad \|A^k\| < \gamma \quad \forall k \geq N \quad (\text{C.25})$$

Furthermore as  $0 \in \text{int}(\mathbb{Y})$  and  $A$  is asymptotically stable lemma C.2.17 and lemma C.2.13 give

$$\exists \gamma > 0 : \quad \mathbb{B}(0, \gamma r) \subset \mathbb{O}_\infty(A, C, \mathbb{Y}) \subset \mathbb{O}_k(A, C, \mathbb{Y}) \quad (\text{C.26})$$

Combining (C.25) and (C.26) gives

$$\exists N \in \mathbb{I} : \quad \mathbb{B}(0, \|A^k\| r) \subset \mathbb{O}_k(A, C, \mathbb{Y}) \quad \forall k \geq N \quad (\text{C.27})$$

Consequently

$$\begin{aligned} & \exists N \in \mathbb{I} \quad \forall k \in \{k \in \mathbb{I} \mid k \geq \max(N, n-1)\} : \\ & x \in \mathbb{O}_k(A, C, \mathbb{Y}) \subset \mathbb{B}(0, r) \quad \Rightarrow \quad A^{k+1}x \in \mathbb{B}(0, \|A^{k+1}\| r) \subset \mathbb{O}_{k+1}(A, C, \mathbb{Y}) \end{aligned} \quad (\text{C.28})$$

This result implies that  $\mathbb{O}_k(A, C, \mathbb{Y}) \subset \mathbb{O}_{k+1}(A, C, \mathbb{Y})$ , which completes the proof.  $\square$

**Remark C.2.19**

For the output sets  $\mathbb{Y}$  considered,  $0 \in \text{int}(\mathbb{Y})$  implies  $y_{\min} < 0 < y_{\max}$ .

**Remark C.2.20**

The  $k$  used in the proof is an upper bound for the output admissibility index,  $t^*$ . Choose  $\gamma > 0$  and  $0 < r < \infty$  such that  $\mathbb{B}(0, \gamma) \subset \mathbb{Y} \subset \mathbb{B}(0, r)$ . Then  $\|CA^{k+1}\| r < \gamma$  implies that  $CA^{k+1}\mathbb{B}(0, r) \in \mathbb{Y}$  and  $t^* \leq k$ .



In a linear model predictive control context, the requirement of asymptotic stability does not limit the application, as the  $A$ -matrix is a closed loop  $A$ -matrix asymptotically stable by construction. Furthermore, as the next proceedings will show the output admissible set of unstable  $A$ -matrices are readily computed, leaving only potential problems with stable  $A$ -matrices, i.e. matrices with simple eigenvalues on the unit circle. For practical linear MPC problems they seldom present a problem, as other characteristics of the solution often provides the necessary requirements to characterize the admissible states.

The observability requirement is serious and not met by all systems. However, the next proceedings will show that it can be circumvented, as the output admissible set of similar systems are related and the maximum output admissible set of unobservable subsystems is easily characterized. Therefore for linear model predictive applications it is no real restriction.

The requirement that  $\mathbb{Y}$  is bounded is not met in all linear model predictive control formulations. Though it is easy to meet the requirement by imposing very large bounds on the unbounded outputs. From a practical point of view this does not lead to performance degradation and is therefore acceptable. From a strict mathematical point of view, the maximal output admissible set, of the modified problem, is a subset of the maximal output admissible set of the original problem. Therefore in the rigorous sense the set found in finite time is only a positively invariant set of the original unbounded problem, but not the maximal output admissible set of that problem. As the practical examples show, this is not a serious limitation of the method.

The last requirement saying that origo must be in the interior of the output constraints profoundly impacts the application of the theorem in a linear model predictive control context, as it means that no constraints can be active at steady state. Loeblein and Perkins (1999a) use the arguments of Narraway *et al.* (1991) to advocate backing off from the constraints active at steady state. Using this approach will fulfill the interior requirement. Rao and Rawlings (1999) address the problem by considering a system projected into the null space of the steady state active constraints. We will later demonstrate the implications of origo being on the boundary of the output admissible set and discuss its linear model predictive control implications.

### C.3 Algorithmic Determination

Theorem C.2.15 prescribes a conceptual method of constructing the maximal output admissible set, and theorem C.3.7 gives sufficient conditions for termination of the conceptual algorithm in finite time.

To make the procedure applicable to practical linear model predictive control computations, we will demonstrate how to deal with the observability requirement and how to test whether  $\mathbb{O}_{t+1}(A, C, \mathbb{Y}) = \mathbb{O}_t(A, C, \mathbb{Y})$ . While not relevant

to current linear model predictive control formulations, we will also show how to deal with the asymptotic stability condition.

The following theorem is the fundamental result used to challenge the observability and asymptotic stability requirements.

**Theorem C.3.1 (The Output Admissible Set of Similar Systems)**

Let  $(A, C, \mathbb{Y})$  and  $(\bar{A}, \bar{C}, \mathbb{Y})$  be similar systems related by  $\bar{x} = Px$  where  $P$  is non-singular and

$$\bar{A} = PAP^{-1} \quad (\text{C.29})$$

$$\bar{C} = CP^{-1} \quad (\text{C.30})$$

The output admissible sets of these systems are related by

$$\mathbb{O}_\infty(A, C, \mathbb{Y}) = P^{-1}\mathbb{O}_\infty(\bar{A}, \bar{C}, \mathbb{Y}) \quad (\text{C.31})$$

*Proof.*

$$\begin{aligned} \mathbb{O}_\infty(A, C, \mathbb{Y}) &= \{x \in \mathbb{R}^n \mid CA^k x \in \mathbb{Y} \quad k \in \mathbb{I}\} \\ &= \{P^{-1}Px \in \mathbb{R}^n \mid CP^{-1}P(AP^{-1}P)^k x \in \mathbb{Y} \quad k \in \mathbb{I}\} \\ &= \{P^{-1}\bar{x} \in \mathbb{R}^n \mid \bar{C}\bar{A}^k \bar{x} \in \mathbb{Y} \quad k \in \mathbb{I}\} \\ &= P^{-1}\mathbb{O}_\infty(\bar{A}, \bar{C}, \mathbb{Y}) \end{aligned}$$

□

Thus by theorem C.3.1 we shall feel free to study the output admissible set construction in whatever coordinate system is most convenient. This has immediate implications as any system can be decomposed into an observable and unobservable subspace as well as decomposed into a subspace that is asymptotic stable and one that is not.

The following theorem handles the observability requirement.

**Theorem C.3.2 (The Output Admissible Set of an Observable/Unobservable System)**

Consider the system

$$\begin{pmatrix} x_o \\ x_{\bar{o}} \end{pmatrix}_{k+1} = \begin{pmatrix} A_{11} & 0 \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} x_o \\ x_{\bar{o}} \end{pmatrix}_k \quad (\text{C.32})$$

$$y_k = (C_1 \quad 0) \begin{pmatrix} x_o \\ x_{\bar{o}} \end{pmatrix}_k \quad (\text{C.33})$$

where  $x_o$  is output observable states, i.e.  $(A_{11}, C_1)$  is observable,  $x_{\bar{o}}$  is output unobservable states,  $\dim(x_o) = n_1$ , and  $\dim(x_{\bar{o}}) = n_2$ .

The output admissible set of this system is

$$\mathbb{O}_\infty(A, C, \mathbb{Y}) = \mathbb{O}_\infty(A_{11}, C_1, \mathbb{Y}) \times \mathbb{R}^{n_2} \quad (\text{C.34})$$

*Proof.*

The structure of the system implies

$$\begin{aligned}(x_o)_{k+1} &= A_{11}(x_o)_k \\ (x_{\bar{o}})_{k+1} &= A_{21}(x_o)_k + A_{22}(x_{\bar{o}})_k \\ y_k &= C_1(x_o)_k\end{aligned}$$

Consequently

$$y_k = CA^k x_0 = C_1 A_{11}^k (x_o)_0$$

$$\begin{aligned}\mathbb{O}_\infty(A, C, \mathbb{Y}) &= \{x \in \mathbb{R}^n \mid CA^k x \in \mathbb{Y} \quad k \in \mathbb{I}\} \\ &= \{(x_o, x_{\bar{o}}) \in \mathbb{R}^{n_1 \times n_2} \mid C_1 A_{11}^k x_o \in \mathbb{Y} \quad k \in \mathbb{I}\} \\ &= \{x_o \in \mathbb{R}^{n_1} \mid C_1 A_{11}^k x_o \in \mathbb{Y} \quad k \in \mathbb{I}\} \times \{x_{\bar{o}} \in \mathbb{R}^{n_2}\} \\ &= \mathbb{O}_\infty(A, C, \mathbb{Y}) \times \mathbb{R}^{n_2}\end{aligned}$$

□

In current linear model predictive control parameterizations,  $A$  is asymptotically stable as it a closed loop matrix. Consequently, assuming  $\mathbb{Y}$  is bounded and  $0 \in \text{int}(\mathbb{Y})$ , the maximum output admissible set may found by performing an observability decomposition (c.f. Chen, 1999) and then applying the algorithm for determination of the maximal output admissible set to the observable subspace only. The resulting maximal output admissible set is constructed by subsequent use of theorems C.3.2 and C.3.1.

By techniques similarly to the techniques applied in theorem C.3.2, the asymptotic stability requirement can be addressed.

### **Theorem C.3.3 (The Output Admissible Set of Unstable Modes)**

Assume

1.  $(A, C)$  is observable
2.  $\mathbb{Y}$  is bounded
3.  $0 \in \mathbb{Y}$

and that the system  $(A, C)$  is partitioned into its stable and unstable subspace according to

$$\begin{pmatrix} x_s \\ x_u \end{pmatrix}_{k+1} = \begin{pmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{pmatrix} \begin{pmatrix} x_s \\ x_u \end{pmatrix}_k \quad (\text{C.35})$$

$$y_k = \begin{pmatrix} C_1 & C_2 \end{pmatrix} \begin{pmatrix} x_s \\ x_u \end{pmatrix}_k \quad (\text{C.36})$$

where  $x_s$  denotes the states in the stable subspace and  $x_u$  denotes the states in the unstable subspace.  $n_1 = \dim(x_s)$  and  $n_2 = \dim(x_u)$ .

Then the maximal output admissible set is given by

$$\mathbb{O}_\infty(A, C, \mathbb{Y}) = \mathbb{O}_\infty(A_{11}, C_1, \mathbb{Y}) \times \{0\}^{n_2} \quad (\text{C.37})$$

*Proof.* See Gilbert and Tan (1991).  $\square$

**Remark C.3.4**

The decomposition into a stable and unstable subspace of the form (C.35)-(C.36) may be obtained by a Schur decomposition (c.f. Golub and Van Loan, 1996). The observability requirement can be met by only applying the Schur decomposition to the observable subspace of the original system.

**Remark C.3.5**

By the form of the system  $A_{11}$  must be stable (in the Lyapunov sense, i.e. have eigenvalues within the unit circle only, and the eigenvalues on the unit circle must be simple), and  $A_{22}$  must have no stable modes, i.e. all eigenvalues must either be outside the unit circle or non-simple eigenvalues on the unit circle.

In current linear model predictive parameterizations there is no need for theorem C.3.3. However, the parameterization suggested by Muske and Rawlings (1993a) has an unstable  $A$  if the model is unstable. In this parameterization, theorem C.3.3 is employed to construct the maximum output admissible set.

The remaining question to be addressed to have a practical algorithm for construction of maximal output admissible sets for polyhedral output constraints of linear systems is the comparison of  $\mathbb{O}_{t+1}(A, C, \mathbb{Y})$  and  $\mathbb{O}_t(A, C, \mathbb{Y})$ . The following sequence of theorems will solve this issue. However, let us emphasize that the results hinges on the fact that  $\mathbb{Y}$  is polyhedral, which fortunately is the case in linear model predictive control applications.

The next theorem gives the principles for construction of the maximum output admissible set of polyhedral output constraints.

**Theorem C.3.6**

Assume that  $A$  is stable and  $\mathbb{Y} = \{y \in \mathbb{R}^p \mid y_{\min} \leq y \leq y_{\max}\}$  then

1.  $g_i : \mathbb{R}^n \mapsto \mathbb{R}$  given by  $g_i(x) = \sup \{e_i^T C A^k x \mid k \in \mathbb{I}\}$  is defined
2.  $h_i : \mathbb{R}^n \mapsto \mathbb{R}$  given by  $h_i(x) = \inf \{e_i^T C A^k x \mid k \in \mathbb{I}\}$  is defined
3.  $\mathbb{O}_\infty(A, C, \mathbb{Y}) = \{x \in \mathbb{R}^n \mid (y_{\min})_i \leq h_i(x) \wedge g_i(x) \leq (y_{\max})_i \quad i \in \{1, \dots, p\}\}$

*Proof.* The proves of 1. and 2. are identical and regards the existence of a supremum and infimum respectively. The Weierstrass theorem (c.f. Luenberger, 1969) is obviously selected to address this question.

First notice that the function  $f_i : \mathbb{R}^p \mapsto \mathbb{R}$  defined by

$$f_i(y) = e_i^T y \quad (\text{C.38})$$

is continuous.

By the stability of  $A$

$$\exists \gamma > 0 \forall x \in \mathbb{R}^n \forall k \in \mathbb{I} : \|y_k\| = \|CA^k x\| < \gamma \|x\| \quad (\text{C.39})$$

Consequently

$$y_k = CA^k x \in \mathbb{B}(0, \gamma \|x\|) \subset \bar{\mathbb{B}}(0, \gamma \|x\|) \quad (\text{C.40})$$

where  $\bar{\mathbb{B}}$  is the closure of  $\mathbb{B}$ . As  $\bar{\mathbb{B}}$  is bounded, closed and finite dimensional it is also compact.

As  $f_i(y)$  is continuous and  $y$  belongs to a compact set  $\bar{\mathbb{B}}$ ,  $g_i(x)$  defined by

$$g_i(x) = \sup \{e_i^T CA^k x \mid k \in \mathbb{I}\} \quad (\text{C.41})$$

exists, i.e.  $g_i(x) < \infty$  for  $\|x\| < \infty$ , according to Weierstrass theorem. By similar arguments  $h_i(x)$  exists.

Result 3. is a direct consequence of the definition of the maximal output admissible set,  $\mathbb{Y}$ ,  $g_i(x)$  and  $h_i(x)$ :

$$\begin{aligned} \mathbb{O}_\infty(A, C, \mathbb{Y}) &= \{x \in \mathbb{R}^n \mid y_{\min} \leq CA^k x \leq y_{\max} \quad k \in \mathbb{I}\} \\ &= \{x \in \mathbb{R}^n \mid (y_{\min})_i \leq e_i^T CA^k x \leq (y_{\max})_i \quad k \in \mathbb{I}, i \in \{1, \dots, p\}\} \\ &= \{x \in \mathbb{R}^n \mid (y_{\min})_i \leq h_i(x) \wedge g_i(x) \leq (y_{\max})_i \quad i \in \{1, \dots, p\}\} \end{aligned} \quad (\text{C.42})$$

□

The principles in theorem C.3.6 is combined with theorems C.2.18 and C.2.15 to provide the principles for construction of the maximum output admissible set in finite time.

**Theorem C.3.7 (Constructive Method for Determination of  $\mathbb{O}_\infty$ )**

If

1.  $A$  is asymptotically stable
2.  $(A, C)$  is observable
3.  $\mathbb{Y} = \{y \in \mathbb{R}^p \mid y_{\min} \leq y \leq y_{\max}\}$  where  $y_{\min} \in (-\infty, 0)$  and  $y_{\max} \in (0, \infty)$

then  $\exists t \in \mathbb{I}$  such that

$$\sup_{x \in \mathbb{O}_t(A, C, \mathbb{Y})} \{e_i^T CA^{t+1} x\} \leq (y_{\max})_i \quad i \in \{1, 2, \dots, p\} \quad (\text{C.43})$$

$$\inf_{x \in \mathbb{O}_t(A, C, \mathbb{Y})} \{e_i^T CA^{t+1} x\} \geq (y_{\min})_i \quad i \in \{1, 2, \dots, p\} \quad (\text{C.44})$$

and the maximum output admissible set is given by

$$\mathbb{O}_\infty(A, C, \mathbb{Y}) = \left\{x \in \mathbb{R}^n \mid y_{\min} \leq CA^k x \leq y_{\max} \quad k \in \{0, 1, \dots, t\}\right\} \quad (\text{C.45})$$

The smallest such  $t$  is the output admissibility index,  $t^*$ .

*Proof.* By the assumptions and theorem C.2.15  $\mathbb{O}_\infty(A, C, \mathbb{Y})$  is finitely determined

$$\exists t \in \mathbb{I} : \quad \mathbb{O}_t(A, C, \mathbb{Y}) = \mathbb{O}_{t+1}(A, C, \mathbb{Y}) = \mathbb{O}_\infty(A, C, \mathbb{Y})$$

which implies

$$\mathbb{O}_{t+1}(A, C, \mathbb{Y}) = \{x \in \mathbb{O}_t(A, C, \mathbb{Y}) \mid y_{\min} \leq CA^{t+1}x \leq y_{\max}\} = \mathbb{O}_t(A, C, \mathbb{Y})$$

Consequently

$$\forall x \in \mathbb{O}_t(A, C, \mathbb{Y}) : \quad y_{\min} \leq CA^{t+1}x \leq y_{\max}$$

which is identical to

$$\sup_{x \in \mathbb{O}_t(A, C, \mathbb{Y})} \{e_i^T CA^{t+1}x\} \leq (y_{\max})_i \quad i \in \{1, 2, \dots, p\} \quad (\text{C.46})$$

$$\inf_{x \in \mathbb{O}_t(A, C, \mathbb{Y})} \{e_i^T CA^{t+1}x\} \geq (y_{\min})_i \quad i \in \{1, 2, \dots, p\} \quad (\text{C.47})$$

Conversely, if  $\exists t \in \mathbb{I}$  satisfying (C.46) and (C.47) then  $\mathbb{O}_t(A, C, \mathbb{Y}) \subset \mathbb{O}_{t+1}(A, C, \mathbb{Y})$ . This combined with lemma C.2.13 implies  $\mathbb{O}_t(A, C, \mathbb{Y}) = \mathbb{O}_{t+1}(A, C, \mathbb{Y})$ . Thus according to theorem C.2.15  $\mathbb{O}_t(A, C, \mathbb{Y}) = \mathbb{O}_\infty(A, C, \mathbb{Y})$ .

By definition the output admissibility index  $t^*$  is defined as the smallest  $t$  with the above properties.

Accordingly the maximal output admissible set is

$$\mathbb{O}_\infty(A, C, \mathbb{Y}) = \mathbb{O}_t(A, C, \mathbb{Y}) = \{x \in \mathbb{R}^n \mid y_{\min} \leq CA^k x \leq y_{\max} \quad k \in \{0, 1, \dots, t\}\}$$

□

Theorem C.3.7 gives a constructive method for generation of the maximal output admissible set. The core new result in the theorem is the methodology for comparing  $\mathbb{O}_{t+1}(A, C, \mathbb{Y})$  and  $\mathbb{O}_t(A, C, \mathbb{Y})$  when  $\mathbb{Y}$  is polyhedral. This result is explicitly stated in the following corollary.

**Corollary C.3.8**

Assume  $\mathbb{Y} = \{y \in \mathbb{R}^p \mid y_{\min} \leq y \leq y_{\max}\}$ . Then

$$\mathbb{O}_{t+1}(A, C, \mathbb{Y}) = \mathbb{O}_t(A, C, \mathbb{Y})$$

if and only if

$$\sup_{x \in \mathbb{O}_t(A, C, \mathbb{Y})} \{e_i^T CA^{t+1}x\} \leq (y_{\max})_i \quad i \in \{1, 2, \dots, p\} \quad (\text{C.48})$$

$$\inf_{x \in \mathbb{O}_t(A, C, \mathbb{Y})} \{e_i^T CA^{t+1}x\} \geq (y_{\min})_i \quad i \in \{1, 2, \dots, p\} \quad (\text{C.49})$$

*Proof.* It follows directly by the arguments in the proof of theorem C.3.7.

□

### C.3.1 Algorithm

The procedure for computation of the maximal output admissible set is summarized in algorithm 6. This algorithm computes the output admissibility index,  $t^*$ , by solution of a sequence of linear programs. Knowledge of the output admissibility index,  $t^*$ , allows construction of the maximal output admissible set,  $\mathbb{O}_\infty = \mathbb{O}_\infty(A, C, y_{\min}, y_{\max})$ .

In model predictive control application  $y_{\min}$  and  $y_{\max}$  will be limits that depends on the location of the origin. This implies that they depend on the disturbances. Consequently, in all practical infinity horizon predictive controller, the admissibility index,  $t^*$ , and the maximal output admissible set must be computed online for each control computation by the predictive controller. In some applications this requirement of online computation of the maximal output admissible set can prohibit implementation of a rigorous infinity horizon predictive controller.

---

**Algorithm 6** Output Admissibility Index,  $t^* = t^*(A, C, y_{\min}, y_{\max})$

---

**Require:**  $(A, C)$  observable.  $A$  asymptotic stable.  $y_{\min} \in (-\infty, 0)$  and  $y_{\max} \in (0, \infty)$ .

- 1:  $t = 0$
- 2: **for**  $i = 1$  to  $p$  **do**
- 3:   Solve

$$G_{i,t+1}^* = \max_x e_i^T C A^{t+1} x \quad (C.50)$$

$$s.t. \quad y_{\min} \leq C A^k x \leq y_{\max} \quad k \in \{0, 1, \dots, t\} \quad (C.51)$$

- 4:   **if**  $G_{i,t+1}^* > (y_{\max})_i$  **then**
- 5:      $t = t + 1$
- 6:     Go to 2:
- 7:   **end if**
- 8:   Solve

$$H_{i,t+1}^* = \min_x e_i^T C A^{t+1} x \quad (C.52)$$

$$s.t. \quad y_{\min} \leq C A^k x \leq y_{\max} \quad k \in \{0, 1, \dots, t\} \quad (C.53)$$

- 9:   **if**  $H_{i,t+1}^* < (y_{\min})_i$  **then**
  - 10:      $t = t + 1$
  - 11:     Go to 2:
  - 12:   **end if**
  - 13: **end for**
  - 14:  $t^* = t$
  - 15:  $\mathbb{O}_\infty(A, C, y_{\min}, y_{\max}) = \{x \in \mathbb{R}^n \mid y_{\min} \leq C A^k x \leq y_{\max} \quad k \in \{0, 1, \dots, t^*\}\}$
-

## C.4 Examples

A number of simple two-dimensional examples are provided to illustrate the properties and geometry of the maximal output admissible set.

### C.4.1 Bounded Observable Output Constraints with Origo in the Interior

The first examples are used to illustrate the core algorithm and some generic geometrical features of maximal output admissible sets.

The constraint set used in the examples are

$$-1 \leq x_1 \leq 1 \quad -1 \leq x_2 \leq 1 \quad -0.5 \leq x_1 + x_2 \leq 2$$

which corresponds to  $y_{min} \leq Cx \leq y_{max}$  where

$$y_{min} = \begin{pmatrix} -1 \\ -1 \\ -0.5 \end{pmatrix} \quad C = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix} \quad y_{max} = \begin{pmatrix} 1 \\ 1 \\ 2 \end{pmatrix}$$

The state admissible set corresponding to these constraints is depicted in figure C.1. As can be inferred from both the form of the constraints and the figure the state admissible set is bounded and origo is in the interior.

#### Example C.4.1 (Distinct Positive Real Eigenvalues)

The matrix

$$A = \begin{pmatrix} 1 & -0.21 \\ 1 & 0 \end{pmatrix}$$

has two real distinct eigenvalues:  $\lambda_1 = 0.7$  with associated eigenvector  $v_1 = (0.5735 \quad 0.8192)^T$  and  $\lambda_2 = 0.3$  with associated eigenvector  $v_2 = (0.2873 \quad 0.9578)^T$ . The vector field, the eigenvectors and representative trajectories and the state admissible set is plotted in figure C.2. By inspection it is obvious that not all states in the admissible set stay in that set. The maximum output admissible set is illustrated in the bottom part of figure C.2. The lines defining the maximum output admissible set is only a small fraction of all the constraints included in the representation. The output admissibility index is  $t^* = 2$ .

It can be observed from the plots that the lines corresponding to the constraints tends to be parallel with one of the eigenvectors.

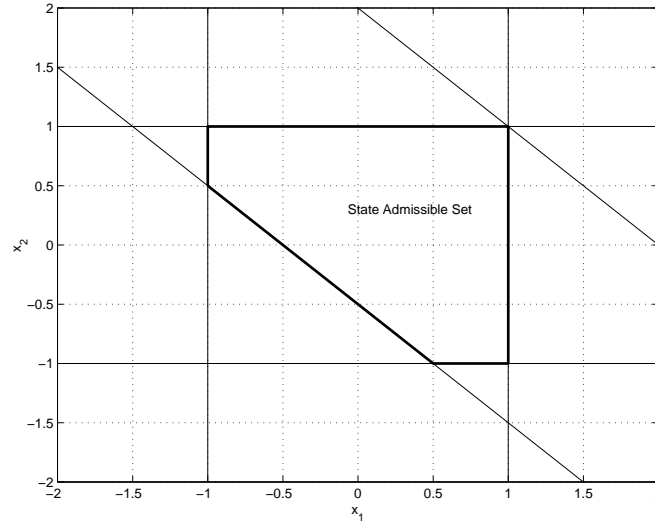
#### Example C.4.2 (A Positive Real and a Negative Real Eigenvalue)

The matrix

$$A = \begin{pmatrix} 0.4 & 0.21 \\ 1 & 0 \end{pmatrix}$$

has a positive real eigenvalue at  $\lambda_1 = 0.7$  with the associated eigenvector  $v_1 = (0.5735 \quad 0.8192)^T$  and a negative real eigenvalue at  $\lambda_2 = -0.3$  with the associated eigenvector  $v_2 = (-0.2873 \quad 0.9578)^T$ .





**Figure C.1.** State admissible set with origo in the interior. The lines correspond to the constraints.

The vector field, the eigenvectors of  $A$ , the state admissible set and representative trajectories are plotted in the upper part of figure C.3. Note that the trajectories cross the eigenvectors. This is possible for discrete time systems with eigenvalues with a negative real part. Therefore for this type of systems the part of the eigenvector lines in the state admissible set is not in general part of the maximum output admissible set. The lower part of figure C.3 shows the maximum output admissible set and the lines defining it. Again it is observed that only a fraction of the equations in the constraint set will be active.

**Example C.4.3 (A Common Positive Real Eigenvalue)**

The matrix

$$A = \begin{pmatrix} 0.5 & 0.3 \\ 0 & 0.5 \end{pmatrix}$$

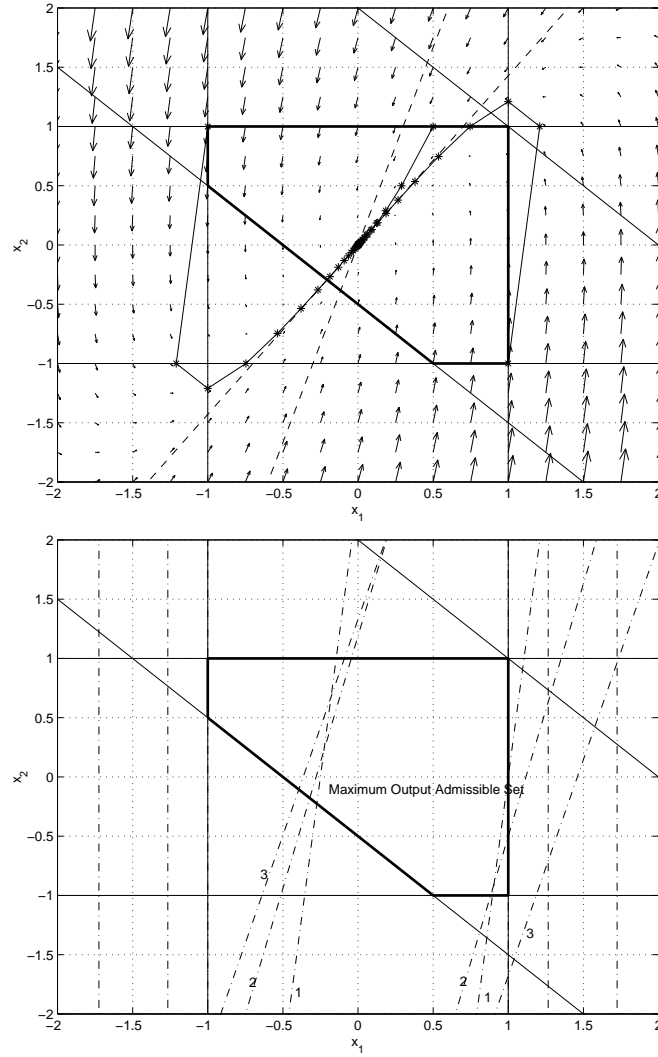
has a double real eigenvalue at  $\lambda = 0.5$ . The corresponding eigenvector is  $v = (1 \ 0)^T$ . The vector field, the eigenvectors, representative trajectories and the state admissible set is illustrated in the upper part of figure C.4. It is evident from the lower part of figure C.4 that almost the entire state admissible set is part of the maximum output admissible set.

**Example C.4.4 (Complex Conjugate Eigenvalues)**

The matrix

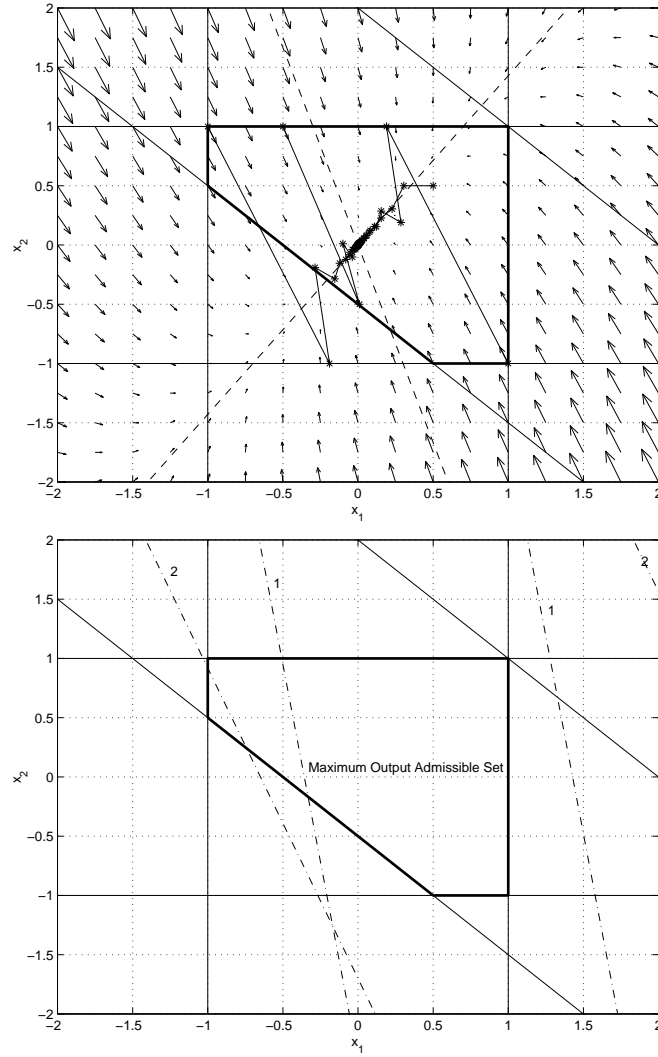
$$A = \begin{pmatrix} 1 & -0.74 \\ 1 & 0 \end{pmatrix}$$

has a complex eigenvalue at  $\lambda = 0.5 \pm 0.7i$ . The vector field and representative trajectories as well as the state admissible set is depicted in the upper part of figure



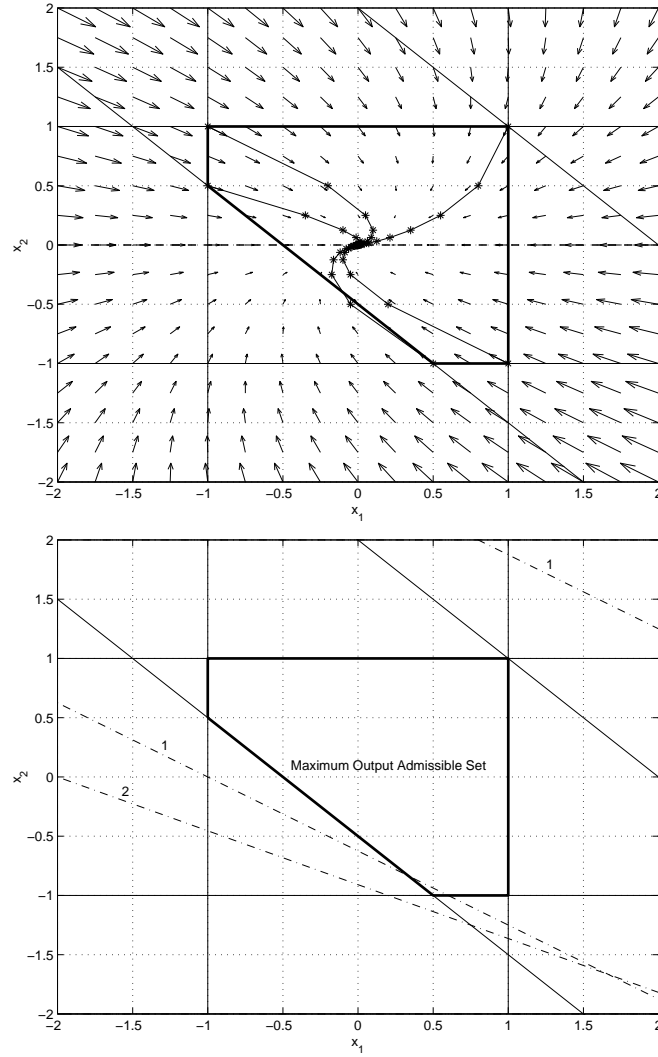
**Figure C.2.** Distinct positive real eigenvalues with origo in the interior. Top: The state admissible set, the eigenvectors of  $A$  and different trajectories. Bottom: Lines corresponding to active constraints at different times and the maximum output admissible set.

C.5. As the eigenvalues of the system are complex conjugate, the vector field forms a spiral. In the bottom part of figure C.5 the lines forming the output constraints  $y_{min} \leq CA^t x \leq y_{max}$  are shown for different values of  $t$ . Due to the complex part of the eigenvalues the output constraints are rotated as  $t$  increases, and due to the stability of the eigenvalue the lines move away from zero in a Minkowski sense. The



**Figure C.3.** Dynamics with a positive real and a negative real eigenvalue. Origo is in the interior of the output constraints. Top: Vector field, eigenvectors and representative trajectories generated by the system. The state admissible set is also shown. Bottom: The maximum output admissible set along with the lines generated by  $y_{min} \leq CA^t x \leq y_{max}$  as function of  $t$ .

maximum output admissible set is indicated in the plot. In this case the active constraints in the output admissible set is  $-0.5 \leq CA^t x$  for  $t \in \{0, 1, 2, 3, 4, 5\}$ . The effect of the vector field on the line  $-0.5 = CA^t x$  is to rotate it outwards. I.e. line 1 can be regarded as the rotation of line 0 by the vector field, line 2 can be regarded



**Figure C.4.** A common positive real eigenvalue and origo in the interior of the output constraint set. Top: Vector field, eigenvector, representative trajectories, and the state admissible set. Bottom: The maximum output admissible set.

as the rotation of line 1 by the vector field, and so on.

The output admissibility index of this system is  $t^* = 5$ . Note that the active constraints defining the maximum output admissible set is a small fraction of all constraints, i.e.  $y_{min} \leq CA^t x \leq y_{max}$  for  $t \in \{0, 1, 2, 3, 4, 5\}$ .

It is important to recognize that the case with complex conjugate eigenvalues is the

generic case for linear model predictive control. The other cases do occur, but the case with complex conjugate eigenvalues is the most common situation.

### C.4.2 Partly Unobservable Output Constraints

The next three examples are used to demonstrate the shape of the maximal output admissible set when the outputs are not observable. The first example is unobservable in the cartesian coordinate system, while the second example is unobservable in a rotated cartesian basis. These two systems both have a common positive real eigenvalue. The third example is used to demonstrate that the results obtained do not depend on the fact that the two first systems only have one real eigenvalue with one associated eigenvector. The last example has two real eigenvalues and two associated eigenvectors.

#### Example C.4.5 (Unobservable System in a Cartesian Basis)

Let the system be

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}_{k+1} = \begin{pmatrix} 0.5 & 0 \\ 0.3 & 0.5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}_k$$

$$-1 \leq \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}_k \leq 1$$

The second state,  $x_2$ , is not observable in the output constraints. The state admissible set is therefore of a cylinder form as indicated in figure C.6. Note that the eigenvector is parallel to the output constraints. This is due to the fact that the system is not observable in the output constraints, i.e. as the output constraint lines and the eigenvector line are parallel they never intersect. In computing the maximum output admissible set is therefore sufficient to consider the dynamics of the first coordinate,  $x_1$ , only. In this case this is trivial as the corresponding eigenvalue is real and stable. Consequently, the output constraints move outward and the maximum output admissible set is identical to the state admissible set. This is illustrated in figure C.6.

The next example is essentially identical to the previous example. The only difference is that the basis is different, i.e. the coordinate system has been rotated.

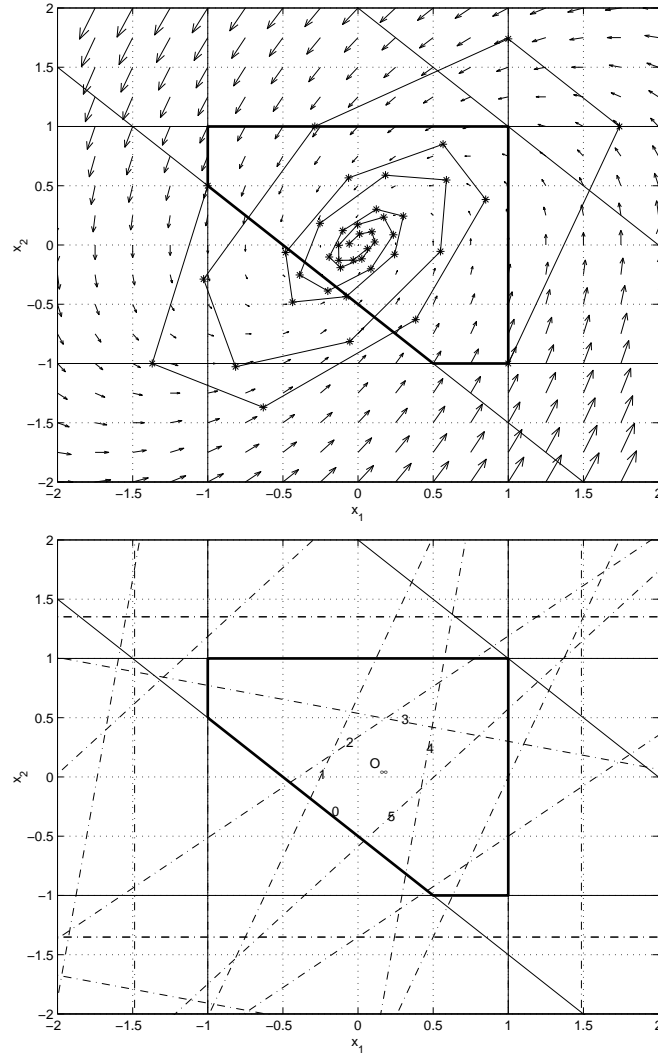
#### Example C.4.6

Let the system be

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}_{k+1} = \begin{pmatrix} 0.35 & -0.15 \\ 0.15 & 0.65 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}_k$$

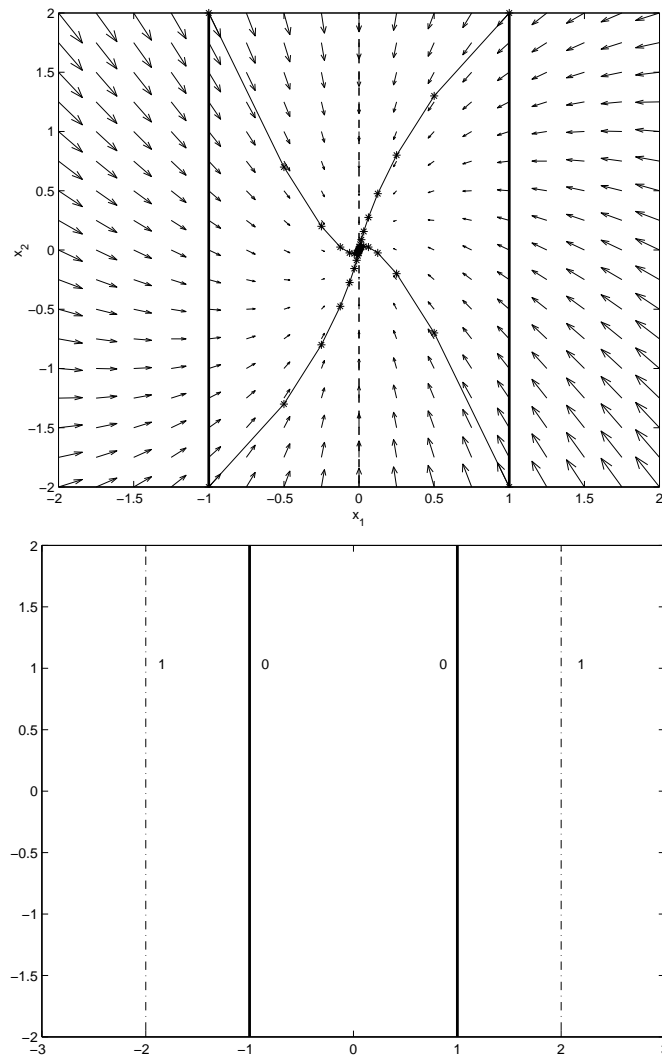
$$-1 \leq \begin{pmatrix} 0.5 & 0.5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}_k \leq 1$$

This system has an eigenvalue at  $\lambda = 0.5$  with multiplicity 2. Just by inspection, it is not obvious that this system is not observable in the output constraints. However



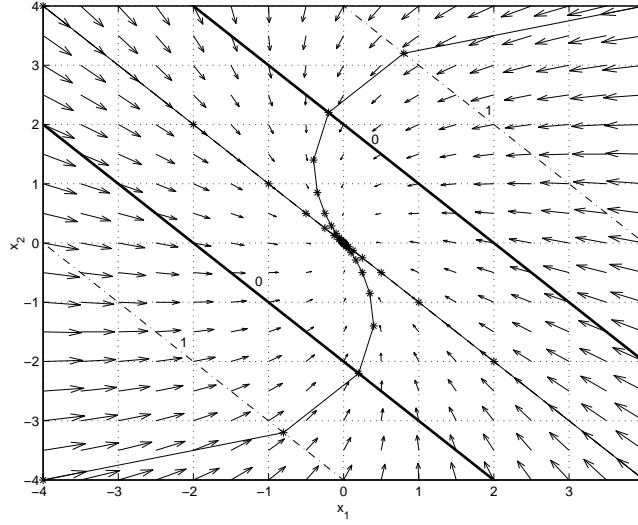
**Figure C.5.** A system with complex conjugate eigenvalues and origo in the interior of the output constraint set. Top: The vector field and trajectories of the system. The state admissible set is also marked. Bottom: The output constraint lines  $y_{min} \leq CA^t x \leq y_{max}$  for different values of  $t$ . The maximum output admissible set is indicated and is relative small subset of the state admissible set.

a Kalman observability decomposition reveals that it is unobservable in the direction  $(1 \ -1)^T$ . This is also revealed in figure C.7 where the eigenvector  $v = (\sqrt{2} \ -\sqrt{2})^T$  is parallel to the output constraints. The output constraints  $y_{min} \leq CA^T x \leq y_{max}$  move outwards parallel to the original constraints as  $t$  increases. The maximum



**Figure C.6.** Unobservable system in the output constraint set. Top: Vector field, eigenvector and representative trajectories. The state admissible set is a cylinder set and indicated. Bottom: The maximum output admissible set is identical to the state admissible set as the constraints move outward (note that the  $x$ -axis are different scaled than in the top).

output admissible set for this system is therefore identical to the state constraint set, which is a cylinder set as indicated in figure C.7.



**Figure C.7.** Unobservable system in the output constraints. The system has a double eigenvalue in 0.5 and therefore only one associated eigenvector. The system is unobservable in the direction of this eigenvector which is parallel to the output constraints.

#### Example C.4.7

The system

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}_{k+1} = \begin{pmatrix} 0.45 & -0.25 \\ 0.05 & 0.75 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}_k$$

$$-1 \leq (0.5 \quad 0.5) \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}_k \leq 1$$

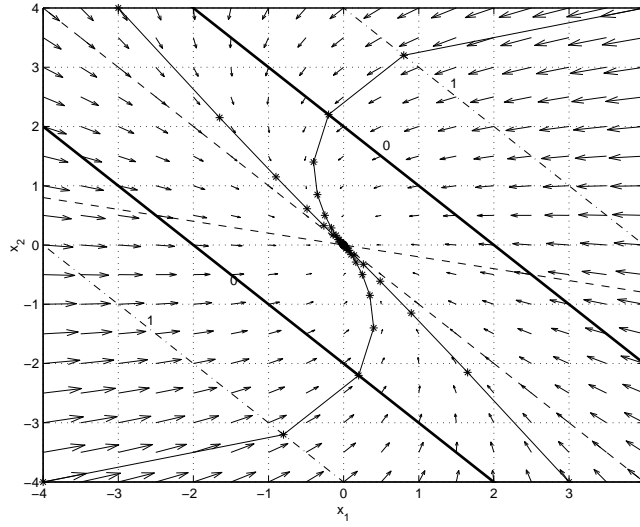
has eigenvalues at  $\lambda_1 = 0.5$  and  $\lambda_2 = 0.7$  with corresponding eigenvectors  $v_1 = (-0.9806 \quad 0.1961)^T$  and  $v_2 = (\sqrt{2} \quad -\sqrt{2})^T$ . Furthermore a Kalman observability decomposition reveals that the system is not observable in the output constraints along the direction  $(1 \quad -1)^T$ . As is illustrated in figure C.8 one of the eigenvectors is parallel with the output constraints. The corresponding mode of the system is therefore not observable.

#### C.4.3 Origo on the Boundary of The Constraint Set

The next examples are used to geometrically demonstrate why the maximal output admissible set is not finitely determined, when origo is on the boundary of the state admissible set. Let the constraints be

$$-1 \leq x_1 \leq 1 \quad -1 \leq x_2 \leq 1 \quad 0 \leq x_1 + x_2 \leq 2$$





**Figure C.8.** Unobservable system in the output constraints. The system has two distinct real eigenvalues and associated distinct eigenvectors. One eigenvector is parallel to the output constraints. The corresponding mode is therefore unobservable and the maximum output admissible set is a cylinder set.

corresponding to  $y_{min} \leq Cx \leq y_{max}$  where

$$y_{min} = \begin{pmatrix} -1 \\ -1 \\ 0 \end{pmatrix} \quad C = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix} \quad y_{max} = \begin{pmatrix} 1 \\ 1 \\ 2 \end{pmatrix} \quad x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

The boundary lines as well as the state admissible set corresponding to these constraints are illustrated in figure C.9.

The maximal output admissible set for this set of output constraints is investigated for dynamic systems

$$x_{k+1} = Ax_k$$

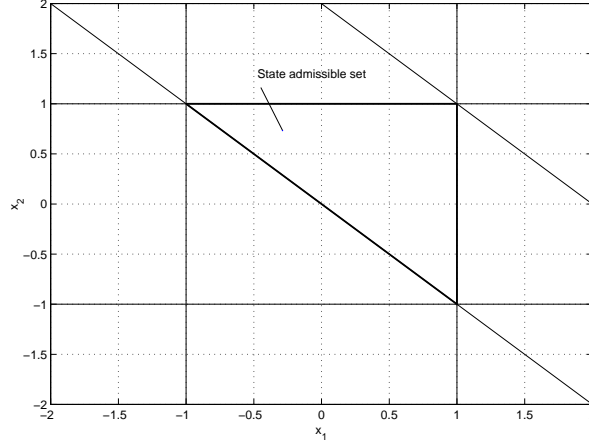
with different  $A$ -matrices.

**Example C.4.8 (Real Distinct Eigenvalues)**

The matrix

$$A = \begin{pmatrix} 1 & -0.21 \\ 1 & 0 \end{pmatrix}$$

has two real distinct eigenvalues:  $\lambda_1 = 0.7$  with associated eigenvector  $v_1 = (0.5735 \quad 0.8192)^T$  and  $\lambda_2 = 0.3$  with associated eigenvector  $v_2 = (0.2873 \quad 0.9578)^T$ . The vector flow, the eigenvector and the constraints are plotted in figure C.10. By inspection of the vector field of this figure it is obvious that not all states in the state admissible set



**Figure C.9.** State admissible set with origo on the boundary. The lines corresponds to the constraints.

belong to the output admissible plot. The output admissible set for different values of  $t$  is also indicated. The plots indicate that the constraint lines  $y_{min} \leq CA^t x \leq y_{max}$  rotate such that they are parallel to the eigenvector with the smallest eigenvalue in the limit  $t \rightarrow \infty$ .

The problem with origo being at the boundary is also illustrated by the lower plot in figure C.10. While the other lines move away from origo in a Minkowski sense, the line through origo keeps going through origo and only rotates. It only approaches the eigenvector in the limit  $t \rightarrow \infty$  and thus the maximal output admissible set is not finitely determined.

**Example C.4.9 (A Positive Real and Negative Real Eigenvalue)**

The matrix

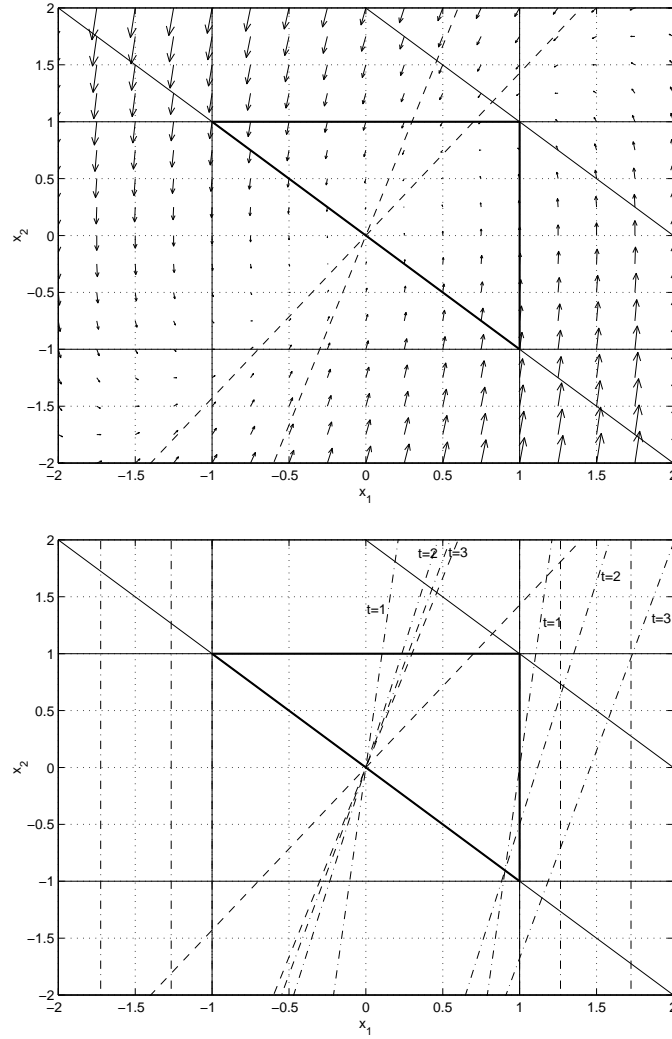
$$A = \begin{pmatrix} 0.4 & 0.21 \\ 1 & 0 \end{pmatrix}$$

has a positive real eigenvalue at  $\lambda_1 = 0$ . with the associated eigenvector  $v_1 = (0.5735 \ 0.8192)^T$  and a negative real eigenvalue at  $\lambda_2 = -0.3$  with the associated eigenvector  $v_2 = (-0.2873 \ 0.9578)^T$ .

The vector field, the eigenvectors (dashed), the constraints (solid), and the constraints (dash-dot) in  $\mathbb{O}_t(A, C, \mathbb{Y})$  are shown in figure C.11.

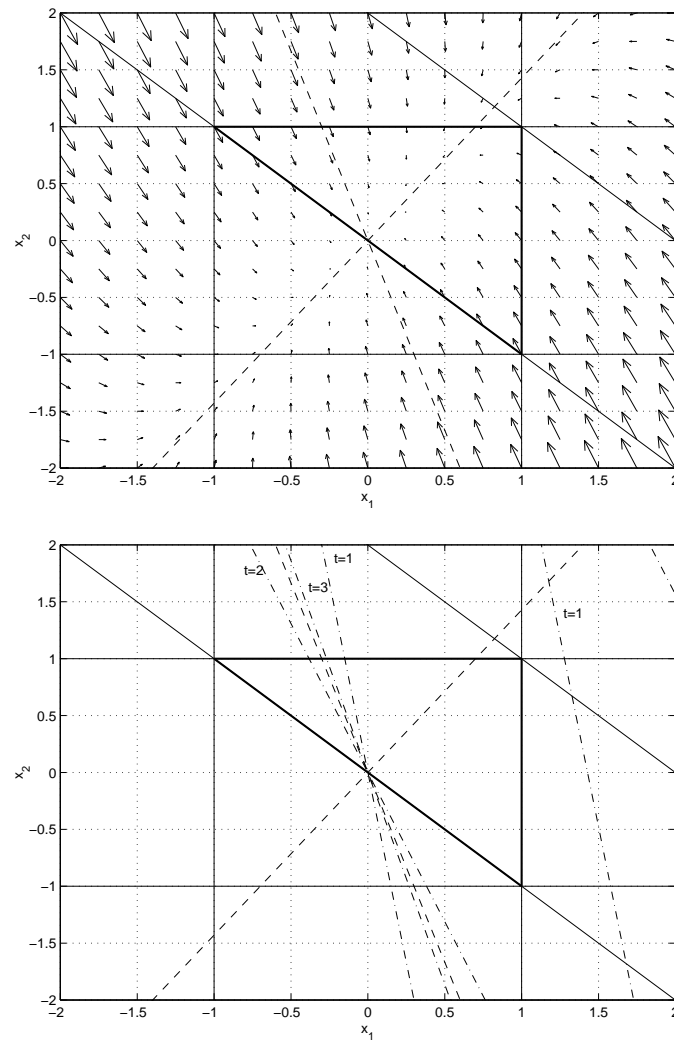
In this case the set produced by the algorithm is finitely determined even though origo is on the boundary of the state admissible set.

In practice discrete-time eigenvalues with negative real value seldom occurs, as the systems observed in engineering typically are sampled continuous-time systems for



**Figure C.10.** Dynamics of a system with two distinct positive real eigenvalues and origo on the boundary of the state admissible set. Top: The vector field, the eigenvectors and the state admissible set. Bottom: The output constraints  $y_{min} \leq CA^t x \leq y_{max}$  for different values of  $t$  and the maximum output admissible set. The constraint through origo approaches one of the eigenvalue lines as  $t \rightarrow \infty$ . Therefore the maximum output admissible set is not finitely determined.

which the discrete-time eigenvalues. For these systems the discrete-time eigenvalues are in the positive half-plane as  $e^{\lambda_i T_s} > 0$ .



**Figure C.11.** A positive real and a negative real eigenvalue. Origo is on the boundary of the state admissible set. Top: The vector field and the eigenvectors as well as the state admissible set. Bottom: The constraints  $y_{min} \leq CA^t x \leq y_{max}$  for different values of  $t$ . Note that the points at lines with  $t = 1$  will be transferred to the output constraints in one time step and so on. As the line through origo passes the eigenvector line, the maximum output admissible set is finitely determined.

**Example C.4.10 (Real Eigenvalue of Multiplicity 2)**

The matrix

$$A = \begin{pmatrix} 0.5 & 0.3 \\ 0 & 0.5 \end{pmatrix}$$

has a double real eigenvalue at  $\lambda = 0.5$ . The corresponding eigenvector is  $v = \begin{pmatrix} 1 & 0 \end{pmatrix}^T$ . The eigenvectors and the vector field along with the state admissible set is depicted in the upper part of figure C.12. Note that the vector field on the boundary in the lower part of the state admissible set is directed out of it indicating that these state are not part of the maximum output admissible set.

The lower part of figure C.12 indicates the rotation of the constraint line passing through origo. As  $t$  increases it is rotated in the clockwise direction. In the limit  $t \rightarrow \infty$  this line approaches the eigenvector line. Consequently, the maximum output admissible set is not finitely determined as is evident from figure C.12.

**Example C.4.11 (Complex Conjugate Eigenvalues)**

The matrix

$$A = \begin{pmatrix} 1 & -0.74 \\ 1 & 0 \end{pmatrix}$$

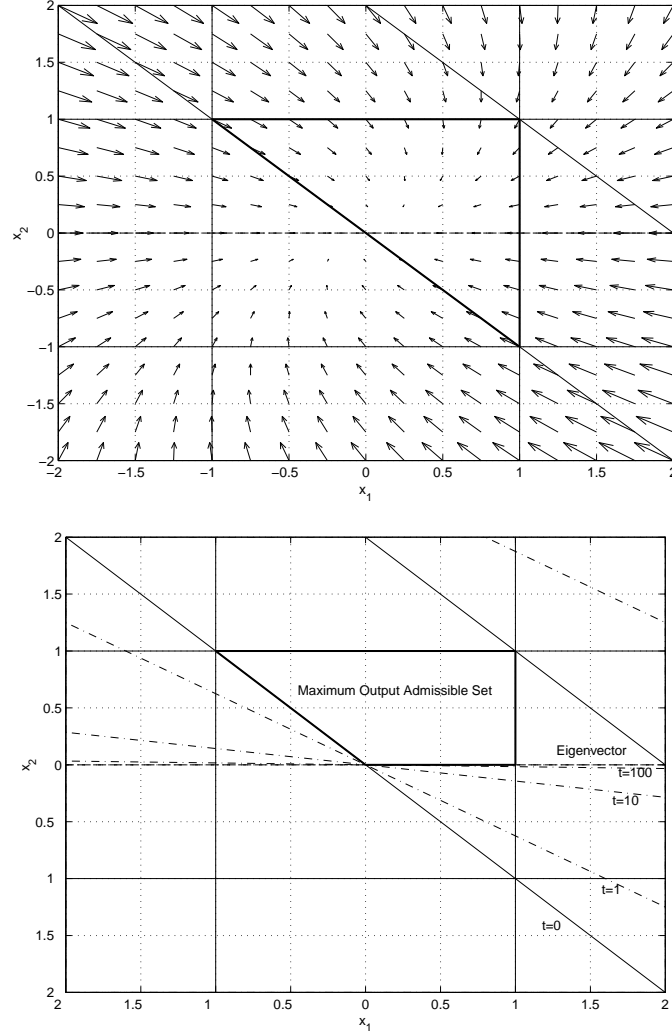
has a complex eigenvalue at  $\lambda = 0.5 \pm 0.7i$ . The vector field of this matrix is shown in figure C.13. Due to the complex eigenvalues the vector field forms a spiral. The stability implies that the constraints  $y_{min} \leq CA^t x \leq y_{max}$  are rotated away from zero in a Minkowski sense as  $t$  increases. The imaginary part gives persistent rotation. As the constraint through origo keeps going through origo and is persistently rotated, the only point in the maximal output admissible set is origo itself. This is indicated in figure C.13.

## C.5 Conclusion

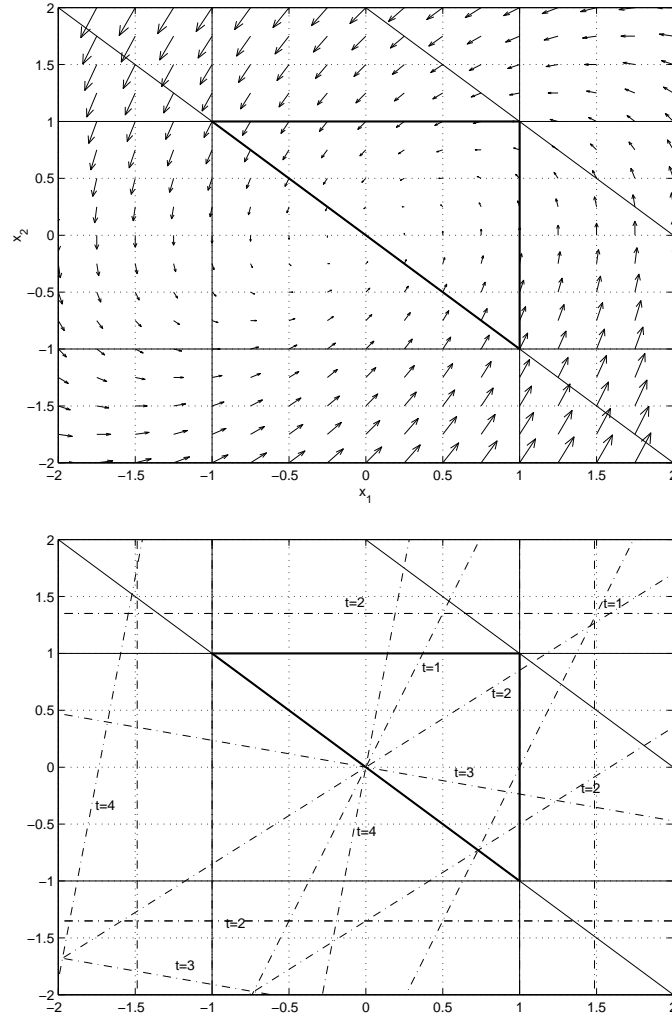
The method of Gilbert and Tan (1991) for construction of the maximal output admissible set for a linear system with polytopic constraints has been presented. The algorithm relies on solution of a sequence of linear programs.

One requirement for termination of the algorithm in a finite number of iterations is that the origin must be in the interior of the constraint set. This requirement prevents application of the algorithm for systems which have active constraints at steady state. Furthermore, in predictive control applications, the origin shifts due to changing disturbances. Hence, the maximal output admissible set must be recomputed online at each control computation.

Consequently, the maximal output admissible set has some limitations in the practical implementation of a true infinite horizon model predictive controller. Nevertheless, the maximal output admissible set serve a useful purpose in study of properties of an ideal infinite horizon constrained predictive controller.



**Figure C.12.** Flow field and eigenvector of the case with a common real eigenvalue at  $\lambda = 0.5$ . Origo is on the boundary of the state admissible set. Top: Vector field and eigenvector line (dashed horizontal line going through origo) and the state admissible set. Bottom: The constraint lines  $y_{min} \leq CA^t x \leq y_{max}$  rotate as function of  $t$ . In the limit  $t \rightarrow \infty$  the constraint line through origo approaches the eigenvector line. The maximum output admissible set is bounded by this eigenvector line and is therefore not finitely determined.



**Figure C.13.** Dynamics of a system with complex conjugate eigenvalues and origo at the boundary of the state admissible set. Top: Flow field and eigenvector of the case with complex eigenvalues. Bottom: Constraints  $y_{min} \leq CA^t x \leq y_{max}$  as function of  $t$ . The vector field rotates the line through origo such that origo is the only point in the maximal output admissible set.

# D

## Linear Regression

Construction of models for predictive control is based on regression. Various regression techniques are reviewed. The regressions described may be applied in an output error as well as a prediction error framework (Söderström and Stoica, 1989; Ljung, 1999).

Draper and Smith (1998) give a general introduction to regression methods. Lütkepohl (1993), Hamilton (1994), and Reinsel (2003) provide comprehensive descriptions of multivariate regression techniques in a multivariate time series analysis context. Reinsel and Velu (1998) describe multivariate reduced rank regression.

### D.1 Regression Framework

The methods for multivariate regression are related to results for the stochastic model

$$\mathbf{y} = \hat{y}(\theta) + \mathbf{v} \quad \mathbf{v} \sim N(0, \Lambda) \quad (\text{D.1})$$

in which  $\theta \in \mathbb{R}^p$ ,  $\mathbf{y} : \Omega \mapsto \mathbb{R}^n$ ,  $\mathbf{v} : \Omega \mapsto \mathbb{R}^n$ . The predictor  $\hat{y} : \mathbb{R}^p \mapsto \mathbb{R}^n$ , is given by a linear relation in the parameters

$$\hat{y}(\theta) = f(\theta) = \Phi\theta \quad (\text{D.2})$$

When the predictor is the deterministic model, the framework is called output error estimation. When the predictor is obtained as either the rigorous or approximate predictor of a stochastic system, the framework is called prediction error estimation. The framework discussed here apply to both situations. The major restriction is that the predictor,  $\hat{y}$ , is assumed to be linear in the parameters,  $\theta$ . Further, the regressor variables  $\Phi$  are assumed to be deterministic.

#### D.1.0.1 Least Squares Estimation

Let  $y \in \mathbb{R}^n$  denote a realization of the stochastic variable  $\mathbf{y}$ . Then the weighted linear least squares estimation problem concerns computing the parameters,  $\theta$ ,



by solution of the problem

$$\min_{\theta \in \mathbb{R}^p} \phi = \frac{1}{2} \|y - \hat{y}(\theta)\|_W^2 = \frac{1}{2} \|y - \Phi\theta\|_W^2 \quad (\text{D.3})$$

It is assumed throughout this section that this problem is strictly convex, i.e. that  $W$  is symmetric positive semi-definite and that  $\Phi'W\Phi$  is positive definite, such that a unique solution exists.

The next results concerns solution of the weighted least squares problem (D.3).

**Proposition D.1.1 (Weighted linear least squares regression)**

The solution of the weighted linear least squares estimation problem

$$\hat{\theta} = \arg \min_{\theta \in \mathbb{R}^p} \phi = \frac{1}{2} \|y - \Phi\theta\|_W^2 \quad (\text{D.4})$$

is

$$\hat{\theta} = (\Phi'W\Phi)^{-1}\Phi'Wy \quad (\text{D.5})$$

*Proof.* The objective function may be expanded

$$\begin{aligned} \phi &= \frac{1}{2} \|y - \Phi\theta\|_W^2 = \frac{1}{2} (y - \Phi\theta)'W(y - \Phi\theta) \\ &= \frac{1}{2} y'Wy - y'W\Phi\theta + \frac{1}{2} \theta'\Phi'W\Phi\theta \\ &= \frac{1}{2} \theta'\Phi'W\Phi\theta - (\Phi'Wy)'\theta + \frac{1}{2} y'Wy \end{aligned} \quad (\text{D.6})$$

As  $\Phi'W\Phi$  is positive definite by assumption, this function is strictly convex. Hence, there exists a unique solution to the problem

$$\hat{\theta} = \arg \min_{\theta \in \mathbb{R}^p} \phi(\theta) \quad (\text{D.7})$$

which is characterized by

$$\nabla \phi(\hat{\theta}) = \Phi'W\Phi\hat{\theta} - \Phi'Wy = 0 \quad (\text{D.8})$$

Consequently

$$\hat{\theta} = (\Phi'W\Phi)^{-1}\Phi'Wy \quad (\text{D.9})$$

□

In the case when  $W = I$ , the weighted linear least squares estimate is called the linear least squares estimate.

**Corollary D.1.2 (Linear least squares estimate)**

The linear least squares estimate

$$\hat{\theta} = \arg \min_{\theta \in \mathbb{R}^p} \phi = \frac{1}{2} \|y - \Phi\theta\|^2 \quad (\text{D.10})$$

is

$$\hat{\theta} = (\Phi'\Phi)^{-1}\Phi'y \quad (\text{D.11})$$

*Proof.* The result is readily obtained from (D.5) using  $W = I$ . □

The minimum variance estimate is a special case of the weighted least squares estimate. In this case, the weights are equal to the inverse of the covariance of the prediction error, i.e.  $W = \Lambda^{-1}$ .

**Corollary D.1.3 (Minimum variance linear least squares estimate)**

The minimum variance estimate

$$\hat{\theta} = \arg \min_{\theta \in \mathbb{R}^p} \phi = \frac{1}{2} \|y - \Phi\theta\|_{\Lambda^{-1}}^2 \quad (\text{D.12})$$

is

$$\hat{\theta} = (\Phi' \Lambda^{-1} \Phi)^{-1} \Phi' \Lambda^{-1} y \quad (\text{D.13})$$

*Proof.* The result follows directly by setting  $W = \Lambda^{-1}$  in (D.5).  $\square$

**D.1.0.2 Maximum Likelihood Estimation**

The likelihood function for the model (D.1)-(D.2) is

$$\begin{aligned} p(y|\theta, \Lambda) &= p(v) = (2\pi)^{-n/2} (\det \Lambda)^{-1/2} \exp \left( -\frac{1}{2} v' \Lambda^{-1} v \right) \\ &= (2\pi)^{-n/2} (\det \Lambda)^{-1/2} \exp \left( -\frac{1}{2} (y - \Phi\theta)' \Lambda^{-1} (y - \Phi\theta) \right) \end{aligned} \quad (\text{D.14})$$

The likelihood function specifies the probability of the measurements given the parameters,  $\theta$  and  $\Lambda$ , in the model (D.1)-(D.2). The maximum likelihood estimator selects the free parameters such that the probability of the measurements is maximized. For cases in which the covariance,  $\Lambda$ , is known the maximum likelihood problem is

$$\max_{\theta \in \mathbb{R}^p} p(y|\theta, \Lambda) \quad (\text{D.15})$$

and for cases in which the covariance is unknown, the maximum likelihood problem is

$$\max_{\theta \in \mathbb{R}^p, \Lambda \in \mathbb{R}^{n \times n}} p(y|\theta, \Lambda) \quad (\text{D.16})$$

It is convenient to introduce the negative logarithm of the likelihood function

$$\begin{aligned} L(\theta, \Lambda; y) &= -\ln p(y|\theta, \Lambda) \\ &= \frac{n}{2} \ln(2\pi) + \frac{1}{2} \ln(\det \Lambda) + \frac{1}{2} (y - \Phi\theta)' \Lambda^{-1} (y - \Phi\theta) \\ &= \frac{n}{2} \ln(2\pi) + \frac{1}{2} \ln(\det \Lambda) + \frac{1}{2} \text{tr}(\Lambda^{-1} (y - \Phi\theta)(y - \Phi\theta)') \end{aligned} \quad (\text{D.17})$$

As the logarithm function,  $\ln$ , is strictly monotonous, the maximum likelihood function with known covariance is equivalent with

$$\min_{\theta \in \mathbb{R}^p} L(\theta, \Lambda; y) \quad (\text{D.18})$$

and the maximum likelihood function without known covariance is equivalent with

$$\min_{\theta \in \mathbb{R}^p, \Lambda \in \mathbb{R}^{n \times n}} L(\theta, \Lambda; y) \quad (\text{D.19})$$

**Proposition D.1.4 (ML estimation with known covariance)**

The maximum likelihood estimate with known covariance,  $\Lambda$ , of the noise

$$\hat{\theta} = \arg \min_{\theta \in \mathbb{R}^p} L(\theta, \Lambda; y) \quad (\text{D.20})$$

is the minimum variance estimate

$$\hat{\theta} = (\Phi' \Lambda^{-1} \Phi)^{-1} \Phi' \Lambda^{-1} y \quad (\text{D.21})$$

*Proof.* As  $\Lambda$  is given, the maximum likelihood problem is

$$\begin{aligned} \min_{\theta \in \mathbb{R}^p} L(\theta, \Lambda; y) &= \frac{n}{2} \ln(2\pi) + \frac{1}{2} \ln(\det \Lambda) + \min_{\theta \in \mathbb{R}^p} \frac{1}{2} (y - \Phi\theta)' \Lambda^{-1} (y - \Phi\theta) \\ &= \frac{n}{2} \ln(2\pi) + \frac{1}{2} \ln(\det \Lambda) + \min_{\theta \in \mathbb{R}^p} \frac{1}{2} \|y - \Phi\theta\|_{\Lambda^{-1}}^2 \end{aligned} \quad (\text{D.22})$$

Hence, the selection of parameters corresponds to a minimum variance linear least squares problem, and the parameters may be computed by (D.13).  $\square$

The next proposition confirms the intuitive notion that we cannot estimate the covariance and the value of the parameters without replicating the experiment, i.e. the maximum likelihood estimate of (D.1)-(D.2) with unknown covariance does not exist. To prove this we use the matrix differentiation stated in the following lemma:

**Lemma D.1.5**

Let  $A \in \mathbb{R}^{n \times n}$  and  $B \in \mathbb{R}^{n \times n}$  be symmetric positive definite matrices. Then

$$\frac{\partial}{\partial A} \ln \det A = A^{-1} \quad (\text{D.23})$$

$$\frac{\partial}{\partial A} \text{tr}(A^{-1}B) = -A^{-1}BA^{-1} \quad (\text{D.24})$$

*Proof.* See Bard (1974).  $\square$

**Proposition D.1.6 (ML estimation with unknown covariance)**

The maximum likelihood estimate

$$(\hat{\theta}, \hat{\Lambda}) = \arg \min_{\theta \in \mathbb{R}^p, \Lambda \in \mathbb{R}^{n \times n}} L(\theta, \Lambda; y) \quad (\text{D.25})$$

of the model (D.1)-(D.2) with unknown covariance,  $\Lambda$ , is undefined and does not exist.

*Proof.* Note that the negative log likelihood function (D.17) is

$$L(\theta, \Lambda; y) = \frac{n}{2} \ln(2\pi) + \frac{1}{2} \ln(\det \Lambda) + \frac{1}{2} \text{tr}(\Lambda^{-1}(y - \Phi\theta)(y - \Phi\theta)') \quad (\text{D.26})$$

To be maximum likelihood estimates,  $(\hat{\theta}, \hat{\Lambda})$  must satisfy the first order optimality conditions

$$\frac{\partial L}{\partial \theta}(\hat{\theta}, \hat{\Lambda}) = 0 \quad (\text{D.27a})$$

$$\frac{\partial L}{\partial \Lambda}(\hat{\theta}, \hat{\Lambda}) = 0 \quad (\text{D.27b})$$

Hence, using lemma D.1.5

$$\frac{\partial}{\partial \Lambda} L(\hat{\theta}, \hat{\Lambda}; y) = \frac{1}{2} \hat{\Lambda}^{-1} - \frac{1}{2} \hat{\Lambda}^{-1}(y - \Phi\hat{\theta})(y - \Phi\hat{\theta})' \hat{\Lambda}^{-1} = 0 \quad (\text{D.28})$$

we obtain

$$\hat{\Lambda} = (y - \Phi\hat{\theta})(y - \Phi\hat{\theta})' \quad (\text{D.29})$$

Consequently

$$\text{tr}(\hat{\Lambda}^{-1}(y - \Phi\hat{\theta})(y - \Phi\hat{\theta})') = \text{tr}(I_n) = n \quad (\text{D.30})$$

and

$$\begin{aligned} L(\hat{\theta}, \hat{\Lambda}; y) &= \frac{n}{2} \ln(2\pi) + \frac{1}{2} \ln(\det \hat{\Lambda}) + \frac{1}{2} n \\ &= \frac{n}{2} [1 + \ln(2\pi)] + \ln(\det[(y - \Phi\hat{\theta})(y - \Phi\hat{\theta})']) \\ &= \frac{n}{2} [1 + \ln(2\pi)] + \ln(\det[\varepsilon\varepsilon']) \end{aligned} \quad (\text{D.31})$$

in which  $\varepsilon = y - \Phi\hat{\theta}$ . The rank of the matrix,  $\varepsilon\varepsilon'$ , is either zero or one. The matrix is of rank 0 when  $\varepsilon = 0$  and of rank one otherwise. Notice that  $y \in \mathbb{R}^n$ . When  $n > 1$ ,  $\det[\varepsilon\varepsilon'] = 0$ , and  $L(\hat{\theta}, \hat{\Lambda}; y)$  is undefined (minus infinity) for all parameters  $\hat{\theta}$ . In the case  $n = 1$  (and  $p \geq 1$ ),  $\varepsilon = 0$  for  $\Phi \neq 0$ . Hence,  $\det[\varepsilon\varepsilon'] = 0$  and  $L(\hat{\theta}, \hat{\Lambda}; y)$  is undefined (minus infinity) for all parameters  $\hat{\theta}$ . The same situation occurs for  $y = 0$  and  $\Phi = 0$ . This leaves the case  $n = 1$ ,  $y \neq 0$  and  $\Phi = 0$ . In this case the covariance exists, but any parameter  $\hat{\theta}$  will give the same likelihood function, and the minima is from a practical point of view undefined in this case too.  $\square$

Not surprisingly, the maximum likelihood estimate of the model (D.1)-(D.2) with unknown covariance,  $\Lambda$ , cannot be computed. In order to compute the covariance, some kind of repetition in the experiments is needed. In section D.2 and D.3, the model is (D.1)-(D.2) specialized to a univariate and multivariate regression model, respectively. These models contain repetition and it is possible to solve the maximum likelihood estimation problem with unknown measurement covariance.

### D.1.0.3 Distribution of the Estimates

#### Proposition D.1.7 (Distribution of the WLS estimate)

The weighted least squares estimate (D.5) of the parameter  $\theta$  in the model (D.1)-(D.2) is normally distributed

$$\hat{\theta} \sim N(\theta, Q) \quad (\text{D.32})$$

with mean  $\theta$  and covariance

$$Q = (\Phi'W\Phi)^{-1}\Phi'W\Lambda W\Phi(\Phi'W\Phi)^{-1} \quad (\text{D.33})$$

*Proof.* The stochastic variable,  $\hat{\theta}$ , is normally distributed as it is a linear combination of a normally distributed variable, i.e.

$$\begin{aligned} \hat{\theta} &= (\Phi'W\Phi)^{-1}\Phi'W\mathbf{y} \\ &= (\Phi'W\Phi)^{-1}\Phi'W(\Phi\theta + \mathbf{v}) \\ &= \theta + (\Phi'W\Phi)^{-1}\Phi'W\mathbf{v} \end{aligned} \quad (\text{D.34})$$

in which  $\mathbf{v} \sim N(0, \Lambda)$ . The mean is

$$E\{\hat{\theta}\} = E\{\theta + (\Phi'W\Phi)^{-1}\Phi'W\mathbf{v}\} = \theta + (\Phi'W\Phi)^{-1}\Phi'WE\{\mathbf{v}\} = \theta \quad (\text{D.35})$$

which implies that the estimate is unbiased. Use that  $W$  is symmetric, to express the covariance of the estimate as

$$\begin{aligned} Q &= V\{\hat{\theta}\} = \langle \hat{\theta} - \theta, \hat{\theta} - \theta \rangle \\ &= \langle (\Phi'W\Phi)^{-1}\Phi'W\mathbf{v}, (\Phi'W\Phi)^{-1}\Phi'W\mathbf{v} \rangle \\ &= [(\Phi'W\Phi)^{-1}\Phi'W] \langle \mathbf{v}, \mathbf{v} \rangle [(\Phi'W\Phi)^{-1}\Phi'W]' \\ &= (\Phi'W\Phi)^{-1}\Phi'W\Lambda W\Phi(\Phi'W\Phi)^{-1} \end{aligned} \quad (\text{D.36})$$

This proves the statement  $\hat{\theta} \sim N(\theta, Q)$ .  $\square$

#### Corollary D.1.8 (Distribution of the LS estimate)

The least squares estimate (D.11) of the parameter  $\theta$  in the model (D.1)-(D.2) is normally distributed

$$\hat{\theta} \sim N(\theta, Q) \quad (\text{D.37})$$

with mean  $\theta$  and covariance

$$Q = (\Phi'\Phi)^{-1}\Phi'\Lambda\Phi(\Phi'\Phi)^{-1} \quad (\text{D.38})$$

*Proof.* This follows directly from proposition D.1.7 using the substitution  $W = I$ .  $\square$

#### Corollary D.1.9 (Distribution of minimum variance estimate)

The minimum variance estimate (D.13) of the parameter  $\theta$  in the model (D.1)-(D.2) is normally distributed

$$\hat{\theta} \sim N(\theta, Q) \quad (\text{D.39})$$

with mean  $\theta$  and covariance

$$Q = (\Phi'\Lambda^{-1}\Phi)^{-1} \quad (\text{D.40})$$

*Proof.* This follows directly from proposition D.1.7 using the substitution  $W = \Lambda^{-1}$ , i.e.

$$\begin{aligned} Q &= (\Phi' \Lambda^{-1} \Phi)^{-1} \Phi' \Lambda^{-1} \Lambda \Lambda^{-1} \Phi (\Phi' \Lambda^{-1} \Phi)^{-1} \\ &= (\Phi' \Lambda^{-1} \Phi)^{-1} \Phi' \Lambda^{-1} \Phi (\Phi' \Lambda^{-1} \Phi)^{-1} \\ &= (\Phi' \Lambda^{-1} \Phi)^{-1} \end{aligned} \quad (\text{D.41})$$

□

**Corollary D.1.10 (Distribution of the ML estimate with known covariance)**

Consider the maximum likelihood estimate with known covariance,  $\Lambda$ , of the noise. The estimate (D.20) of  $\theta$  is normally distributed

$$\hat{\theta} \sim N(\theta, Q) \quad (\text{D.42})$$

with mean  $\theta$  and covariance

$$Q = (\Phi' \Lambda^{-1} \Phi)^{-1} \quad (\text{D.43})$$

*Proof.* This is a trivial consequence of corollary D.1.9, as the maximum likelihood estimate with known noise covariance is identical to the minimum variance least squares estimate. □

### D.1.1 Numerical Methods

The optimum,  $\hat{\theta}$ , of the least squares problem

$$\hat{\theta} = \arg \min_{\theta} \left\{ \phi = \frac{1}{2} \|y - \Phi \theta\|_2^2 \right\} \quad (\text{D.44})$$

satisfies the first order optimality conditions (the normal equations)

$$(\Phi' \Phi) \hat{\theta} = \Phi' y \quad (\text{D.45})$$

and may formally be obtained as

$$\hat{\theta} = (\Phi' \Phi)^{-1} \Phi' y \quad (\text{D.46})$$

Various numerical methods for solution of the normal equations and thereby the least squares problem are presented.

The normal equations (D.45) may be expressed as the linear system of equations

$$A \hat{\theta} = b \quad (\text{D.47a})$$

in which

$$A = \Phi' \Phi \quad (\text{D.47b})$$

$$b = \Phi' y \quad (\text{D.47c})$$

and  $A$  is symmetric positive semi definite. The solution to (D.47a) may be expressed as

$$\hat{\theta} = A^{-1}b \quad (\text{D.48})$$

provided  $A$  is non-singular. It should be noted that (D.47a) can be regarded as the first order optimality conditions for the quadratic program

$$\hat{\theta} = \arg \min_{\theta} \left\{ \tilde{\phi} = \frac{1}{2} \theta' A \theta - b' \theta \right\} \quad (\text{D.49})$$

#### D.1.1.1 LU Factorization of the Normal Equations

The coefficient matrix,  $A$ , associated with the normal equations may be LU factorized

$$A = \Phi' \Phi = LU \quad (\text{D.50})$$

such that the least squares estimate may be computed by

$$\hat{\theta} = A^{-1}b = U^{-1}[L^{-1}(\Phi'y)] \quad (\text{D.51})$$

This factorization is not recommended for solution of the normal equations as it does not utilize the symmetry of  $A$ . Neither, does it provide any mechanism to handle ill-conditioned matrices,  $A$ .

#### D.1.1.2 Cholesky Factorization of the Normal Equations

Provided the coefficient matrix,  $A = \Phi' \Phi$ , is positive definite (non-singular), it may be factorized using a Cholesky factorization

$$A = \Phi' \Phi = LL' \quad (\text{D.52})$$

Using this Cholesky factorization, the optimal estimate of the parameters are obtained as

$$\hat{\theta} = A^{-1}b = (L')^{-1}[L^{-1}(\Phi'y)] \quad (\text{D.53})$$

The Cholesky factorization is computationally more efficient than the LU factorization as it utilize the symmetry of  $A$ . The requirement of positive definiteness implies in practice that  $A$  must be well-conditioned.

#### D.1.1.3 QR Factorization of the Normal Equations

Yet another alternative for solution of the normal equations is by a QR factorization. In this case  $A$  is factorized as

$$A = \Phi' \Phi = Q \begin{bmatrix} R \\ 0 \end{bmatrix} = [Q_1 \quad Q_2] \begin{bmatrix} R \\ 0 \end{bmatrix} = Q_1 R \quad (\text{D.54})$$

in which  $Q$  is an orthogonal matrix ( $Q'Q = QQ' = I$ ) and  $R$  is a non-singular upper (right) triangular matrix. This implies that

$$\hat{\theta} = R^{-1}[Q'_1(\Phi'y)] \quad (\text{D.55})$$

The QR-factorization is computationally more expensive than the Cholesky factorization. However, it has a built in mechanism to handle ill-conditioned systems.

#### D.1.1.4 SVD Factorization of the Normal Equations

The SVD factorization of  $A$  may be stated as

$$A = \Phi'\Phi = U\Sigma V' = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1 & V_2 \end{bmatrix}' = U_1\Sigma_1V_1' \quad (\text{D.56})$$

in which  $U$  and  $V$  are unitary matrices ( $U'U = UU' = I$  and  $V'V = VV' = I$ ) and  $\Sigma$  is a diagonal matrix. Consequently,  $U_1'U_1 = I$ ,  $V_1'V_1 = I$ , and  $\Sigma_1$  is diagonal and non-singular. Using this factorization, the solution of the normal equations is computed as

$$\hat{\theta} = A^{-1}b = V_1 (\Sigma_1^{-1}[U_1'(\Phi'y)]) \quad (\text{D.57})$$

The SVD factorization is even more expensive than the QR factorization. However, it is very efficient for detecting ill-conditioning and may therefore be the preferred alternative for some ill-conditioned systems.

#### D.1.1.5 Recursive Factorization of the Normal Equations

Let  $y = [y_1 \ y_2 \ \dots \ y_n]'$  and  $\theta \in \mathbb{R}^p$ . Furthermore, define  $\varphi_k \in \mathbb{R}^p$  as the columns of  $\Phi'$  ( $\Phi \in \mathbb{R}^{n \times p}$ ), i.e.

$$\Phi' = \Phi(n)' = [\varphi_1 \ \varphi_2 \ \dots \ \varphi_n] \quad (\text{D.58})$$

This implies that

$$\begin{aligned} A(k) &= \Phi(k)'\Phi(k) = \sum_{j=1}^k \varphi_j \varphi_j' \\ &= \sum_{j=1}^{k-1} \varphi_j \varphi_j' + \varphi_k \varphi_k' = A(k-1) + \varphi_k \varphi_k' \end{aligned} \quad (\text{D.59})$$

To compute the least squares estimate using an equation like (D.59), the matrix inversion lemma stated next is needed.



**Lemma D.1.11 (Matrix Inversion Lemma)**

$$(A + BCD)^{-1} = A^{-1} - A^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA^{-1} \quad (\text{D.60})$$

*Proof.* See Kailath *et al.* (2000).  $\square$

Let  $A = A(k)$ ,  $B = \varphi_k$ ,  $C = 1$ , and  $D = \varphi'_k$ . Application of the matrix inversion lemma to (D.59) yields

$$\begin{aligned} A_k^{-1} &= A_{k-1}^{-1} - A_{k-1}^{-1}\varphi_k(1 + \varphi'_k A_{k-1}^{-1}\varphi_k)^{-1}\varphi'_k A_{k-1}^{-1} \\ &= A_{k-1}^{-1} - \frac{(A_{k-1}^{-1}\varphi_k)(A_{k-1}^{-1}\varphi_k)'}{1 + \varphi'_k(A_{k-1}^{-1}\varphi_k)} \end{aligned} \quad (\text{D.61})$$

In the last expression, the symmetry of  $A_{k-1}$  is exploited. Let  $P_k = A_k^{-1}$ . This implies that (D.61) may be expressed as

$$P_k = P_{k-1} - \frac{(P_{k-1}\varphi_k)(P_{k-1}\varphi_k)'}{1 + \varphi'_k(P_{k-1}\varphi_k)} = P_{k-1} - \frac{P_{k-1}\varphi_k\varphi'_k P_{k-1}}{1 + \varphi'_k P_{k-1}\varphi_k} \quad (\text{D.62})$$

Define  $\bar{y}_k = [y_1 \ y_2 \ \dots \ y_k]'$  and the estimate  $\hat{\theta}_k$  as the least square estimate given  $\bar{y}_k$ . Then

$$\begin{aligned} \hat{\theta}_k &= (\Phi(k)' \Phi(k))^{-1} \Phi(k)' \bar{y}_k = P_k [\Phi(k-1)' \ \varphi_k] \begin{bmatrix} \bar{y}_{k-1} \\ y_k \end{bmatrix} \\ &= \left( P_{k-1} - \frac{P_{k-1}\varphi_k\varphi'_k P_{k-1}}{1 + \varphi'_k P_{k-1}\varphi_k} \right) (\Phi(k-1)' \bar{y}_{k-1} + \varphi_k y_k) \\ &= \underbrace{P_{k-1}\Phi(k-1)' \bar{y}_{k-1}}_{=\hat{\theta}_{k-1}} - \frac{P_{k-1}\varphi_k}{1 + \varphi'_k P_{k-1}\varphi_k} \varphi'_k \underbrace{P_{k-1}\Phi(k-1)' \bar{y}_{k-1}}_{=\hat{\theta}_{k-1}} \\ &\quad + P_{k-1}\varphi_k \underbrace{\left( 1 - \frac{\varphi'_k P_{k-1}\varphi_k}{1 + \varphi'_k P_{k-1}\varphi_k} \right)}_{=\frac{1}{1 + \varphi'_k P_{k-1}\varphi_k}} y_k \\ &= \hat{\theta}_{k-1} + \frac{P_{k-1}\varphi_k}{1 + \varphi'_k P_{k-1}\varphi_k} (y_k - \varphi'_k \hat{\theta}_{k-1}) \end{aligned} \quad (\text{D.63})$$

Hence, the recursions for computation of  $\hat{\theta}_k$  are

$$K_k = \frac{P_{k-1}\varphi_k}{1 + \varphi'_k(P_{k-1}\varphi_k)} \quad (\text{D.64a})$$

$$P_k = P_{k-1} - \frac{(P_{k-1}\varphi_k)(P_{k-1}\varphi_k)'}{1 + \varphi'_k(P_{k-1}\varphi_k)} = P_{k-1} - K_k(P_{k-1}\varphi_k)' \quad (\text{D.64b})$$

$$\hat{\theta}_k = \hat{\theta}_{k-1} + K_k(y_k - \varphi'_k \hat{\theta}_{k-1}) \quad (\text{D.64c})$$

and the least squares estimate,  $\hat{\theta}$ , is obtained as  $\hat{\theta}_n$ . The remaining problem concerns selection of  $\hat{\theta}_0$  and  $P_0$  to initiate the recursion. It can be shown that the estimate,  $\hat{\theta}_n$ , corresponds to the solution of the regularized least squares problem

$$\hat{\theta}_n = \arg \min_{\theta} \left\{ \phi = \frac{1}{2} \|y - \Phi\theta\|_2^2 + \frac{1}{2} \|\theta - \theta_0\|_{P_0^{-1}}^2 \right\} \quad (\text{D.65a})$$

One rigorous way to select  $\theta_0 \in \mathbb{R}^p$  and  $P_0 \in \mathbb{R}^{p \times p}$  is to use the first  $p$  data points,  $\bar{y}_p$ , and compute  $\theta_0 = \hat{\theta}_p$  and  $P_0 = (\Phi(p)' \Phi(p))^{-1}$  by a non-recursive least squares procedure based on the first  $p$  data points. In this approach  $\Phi(p)' \Phi(p)$  is assumed to be non-singular. Subsequently, the estimates are computed iteratively disregarding the first  $p$  data points.

Alternatively,  $\theta_0 = 0$  and  $P_0$  is selected heuristically as a large number such that  $P_0^{-1} \approx 0$ . A common choice is  $P_0 = \alpha I$  in which  $\alpha$  is a large number. In this case the second term of (D.65) becomes negligible and (D.65) approximates the least squares problem well.

**Kalman Filter Interpretation:** The recursive equations for solution of the least squares problem may be regarded as a Kalman filter for the least squares problem (Söderström and Stoica, 1989; Sayed and Kailath, 1994; Ljung, 1999; Kailath *et al.*, 2000). The least-squares problem (D.44) with  $\Phi' = [\varphi_k]_{k=1}^N$  may be interpreted as an estimation problem in the dynamic stochastic system

$$\theta_{k+1} = \theta_k \quad (\text{D.66a})$$

$$y_k = \varphi_k' \theta_k + v_k \quad (\text{D.66b})$$

in which  $v_k \sim N(0, 1)$  and  $\theta_0 \sim N(\hat{\theta}_{0|-1}, P_{0|-1})$ . The predictive Kalman filter recursions for the system

$$x_{k+1} = A_k x_k + G w_k \quad (\text{D.67a})$$

$$y_k = C_k x_k + v_k \quad (\text{D.67b})$$

in which

$$\left\langle \begin{bmatrix} w_k \\ v_k \\ x_0 \end{bmatrix}, \begin{bmatrix} w_l \\ v_l \\ x_0 \\ 1 \end{bmatrix} \right\rangle = \begin{bmatrix} Q_k \delta_{kl} & S_k \delta_{kl} & 0 & 0 \\ S_k \delta_{kl} & R_k \delta_{kl} & 0 & 0 \\ 0 & 0 & P_{0|-1} & \hat{x}_{0|-1} \end{bmatrix} \quad (\text{D.67c})$$

are

$$R_{e,k} = R_k + C_k P_{k|k-1} C_k' \quad (\text{D.68a})$$

$$K_{p,k} = (A_k P_{k|k-1} C_k' + G_k S_k) R_{e,k}^{-1} \quad (\text{D.68b})$$

$$P_{k+1|k} = A_k P_{k|k-1} A_k' + G_k Q_k G_k' - K_{p,k} R_{e,k} K_{p,k}' \quad (\text{D.68c})$$

$$\hat{x}_{k+1|k} = A_k \hat{x}_{k|k-1} + K_{p,k} (y_k - C_k \hat{x}_{k|k-1}) \quad (\text{D.68d})$$

Consequently, the predictive Kalman filter for (D.66) becomes

$$R_{e,k} = 1 + \varphi'_k P_{k|k-1} \varphi_k \quad (\text{D.69a})$$

$$K_{p,k} = P_{k|k-1} \varphi_k R_{e,k}^{-1} = \frac{P_{k|k-1} \varphi_k}{1 + \varphi'_k P_{k|k-1} \varphi_k} \quad (\text{D.69b})$$

$$P_{k+1|k} = P_{k|k-1} - K_{p,k} R_{e,k} K'_{p,k} = P_{k|k-1} - K_{p,k} (P_{k|k-1} \varphi_k)' \quad (\text{D.69c})$$

$$\hat{\theta}_{k+1|k} = \hat{\theta}_{k|k-1} + K_{p,k} (y_k - \varphi'_k \hat{\theta}_{k|k-1}) \quad (\text{D.69d})$$

It is apparent that except from some minor notational details, the predictive Kalman filter equations (D.69) are identical to the recursions (D.64) for the recursive least squares solution procedure. This observation is important because it implies that parameter estimation by least squares methods can be expressed as a filtering problem and solved using control, filtering and estimation techniques of dynamic systems (Sayed and Kailath, 1994; Kailath *et al.*, 2000).

#### D.1.1.6 Iterative Solution of the Normal Equations

The iterative method presented for solution of  $A\theta = b$  in which  $A = \Phi'\Phi \in \mathbb{R}^{p \times p}$  is symmetric and positive definite is a conjugate gradient method (c.f. Golub and Van Loan, 1996; Björck, 1996; Nocedal and Wright, 1999) for optimization of (D.49). Minimization of  $\phi = \frac{1}{2}\theta' A\theta - b'\theta$  is equivalent to solution of  $A\theta = b$ . The conjugate gradient algorithm applies at most  $p$  iterations for solution  $A\theta = b$ . The conjugate algorithm can also be adopted for solution of ill-conditioned problems by terminating the algorithms prematurely, i.e. before the residual is zero (Hansen, 1996).

The conjugate gradient (CG) algorithm is useful for large scale systems as it does not need to store or factorize the matrix  $A$  but only need the  $A$  operator, i.e. the matrix-vector multiplication  $Ap_k$  (and  $A\theta_0$ ). If  $A$  is sparse this may easily be utilized as well in this operation. The conjugate gradient algorithm for solution of  $\min_{\theta} \{\phi = \frac{1}{2}\theta' A\theta - b'\theta\}$  equivalent to solution of  $A\theta = b$  is stated in algorithm 7.

The conjugate gradient algorithm (algorithm 7) may be specialized to solution of the normal equations for the least squares problem. This algorithm is called the conjugate gradient least squares (CGLS) algorithm and is stated as algorithm 8. This algorithm solves  $\min_{\theta} \{\phi = \frac{1}{2} \|y - \Phi\theta\|^2\}$  which is equivalent to solution of  $\Phi'\Phi\theta = \Phi'y$  by the conjugate gradient algorithm. The algorithm does not need to store or factorize the matrix,  $\Phi$ . It only need  $\Phi$  as an operator for the matrix vector operations,  $\Phi p_k$  and  $\Phi' \tilde{p}_k$ . It is straightforward to utilize sparsity of  $\Phi$  in these operations and they are also relatively easy to parallelize.

---

**Algorithm 7** CG algorithm for solution of  $A\theta = b$ .

---

**Require:**  $\theta_0, b, A$  symmetric, positive definite.

Set  $r_0 \leftarrow A\theta_0 - b$ ,  $p_0 \leftarrow -r_0$ ,  $\gamma_0 \leftarrow r_0' r_0$ ,  $k \leftarrow 0$ 
**while**  $\gamma_k \neq 0$  **do**

Compute

$$\alpha_k \leftarrow \frac{\gamma_k}{p_k' A p_k} \quad (\text{D.70a})$$

$$\theta_{k+1} \leftarrow \theta_k + \alpha_k p_k \quad (\text{D.70b})$$

$$r_{k+1} \leftarrow r_k + \alpha_k A p_k \quad (\text{D.70c})$$

$$\gamma_{k+1} \leftarrow r_{k+1}' r_{k+1} \quad (\text{D.70d})$$

$$\beta_{k+1} \leftarrow \frac{\gamma_{k+1}}{\gamma_k} \quad (\text{D.70e})$$

$$p_{k+1} \leftarrow -r_{k+1} + \beta_{k+1} p_k \quad (\text{D.70f})$$

$$k \leftarrow k + 1 \quad (\text{D.70g})$$

**end while**
 $\theta \leftarrow \theta_k$ .

---



---

**Algorithm 8** CGLS algorithm for solution of  $\Phi' \Phi \theta = \Phi' y$ .

---

**Require:**  $\theta_0, y, \Phi$ 

Set  $\tilde{r} = \Phi \theta_0 - y$ ,  $r_0 \leftarrow \Phi' \tilde{r}$ ,  $p_0 \leftarrow -r_0$ ,  $\gamma_0 \leftarrow r_0' r_0$ ,  $k \leftarrow 0$ 
**while**  $\gamma_k \neq 0$  **do**

Compute

$$\tilde{p}_k \leftarrow \Phi p_k \quad (\text{D.71a})$$

$$\alpha_k \leftarrow \frac{\gamma_k}{\tilde{p}_k' \tilde{p}_k} \quad (\text{D.71b})$$

$$\theta_{k+1} \leftarrow \theta_k + \alpha_k p_k \quad (\text{D.71c})$$

$$\tilde{r}_k \leftarrow \Phi' \tilde{p}_k \quad (\text{D.71d})$$

$$r_{k+1} \leftarrow r_k + \alpha_k \tilde{r}_k \quad (\text{D.71e})$$

$$\gamma_{k+1} \leftarrow r_{k+1}' r_{k+1} \quad (\text{D.71f})$$

$$\beta_{k+1} \leftarrow \frac{\gamma_{k+1}}{\gamma_k} \quad (\text{D.71g})$$

$$p_{k+1} \leftarrow -r_{k+1} + \beta_{k+1} p_k \quad (\text{D.71h})$$

$$k \leftarrow k + 1 \quad (\text{D.71i})$$

**end while**
 $\theta \leftarrow \theta_k$ .

---

### D.1.1.7 QR Factorization of $\Phi$

The estimates that are computed based on direct solution of the normal equations rely on some factorization of  $A = \Phi'\Phi$ . As  $A$  is the square of  $\Phi$ , its condition number,  $\kappa(A)$ , is the square of the condition number,  $\kappa(\Phi)$ , of  $\Phi$ , i.e.  $\kappa(A) = \kappa(\Phi)^2$ . The precision of the resulting solution is inversely related to the condition number, i.e. the larger a condition number the less precise a solution and vice versa. For ill-conditioned matrices  $\Phi$ , this implies that  $A$  is very ill-conditioned and the resulting solution of the normal equations based on factorization of  $A$  is less accurate than a solution obtained by factorization of  $\Phi$ .

The QR factorization of  $\Phi$  is

$$\Phi = Q \begin{bmatrix} R \\ 0 \end{bmatrix} = [Q_1 \quad Q_2] \begin{bmatrix} R \\ 0 \end{bmatrix} = Q_1 R \quad (\text{D.72})$$

in which  $Q$  is an orthogonal matrix  $Q'Q = QQ' = I$  and  $R$  is a non-singular right triangular matrix. Note the relations

$$\Phi'\Phi = (Q_1 R)'(Q_1 R) = R'Q_1'Q_1 R = R'R \quad (\text{D.73a})$$

$$(\Phi'\Phi)^{-1}\Phi' = (R'R)^{-1}(Q_1 R)' = R^{-1}Q_1' \quad (\text{D.73b})$$

which implies that the least squares estimate may be computed by

$$\hat{\theta} = (\Phi'\Phi)^{-1}\Phi'y = R^{-1}Q_1'y \quad (\text{D.74})$$

### D.1.1.8 SVD Factorization of $\Phi$

$\Phi$  may also be factorized by the SVD factorization

$$\Phi = U\Sigma V' = [U_1 \quad U_2] \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} [V_1 \quad V_2]' = U_1 \Sigma_1 V_1' \quad (\text{D.75})$$

in which  $U$  is unitary ( $U'U = UU' = I$ ),  $V$  is unitary ( $V'V = VV' = I$ ), and  $\Sigma$  is diagonal. Then  $U_1'U_1 = I$ ,  $V_1'V_1 = I$ , and  $\Sigma_1$  is diagonal and non-singular. This implies

$$\Phi'\Phi = V_1 \Sigma_1 U_1' U_1 \Sigma_1 V_1' = V_1 \Sigma_1^2 V_1' \quad (\text{D.76a})$$

$$(\Phi'\Phi)^{-1} = V_1 \Sigma_1^{-2} V_1' \quad (\text{D.76b})$$

$$(\Phi'\Phi)^{-1}\Phi' = V_1 \Sigma_1^{-2} V_1' V_1 \Sigma_1 U_1' = V_1' \Sigma_1^{-1} U_1' \quad (\text{D.76c})$$

Consequently, the least squares estimate may be computed by

$$\hat{\theta} = (\Phi'\Phi)^{-1}\Phi'y = V_1 \Sigma_1^{-1} U_1' y \quad (\text{D.77})$$

## D.2 Univariate Regression

An important special regression problem is the *linear univariate Gaussian* regression problem. The linear multivariate Gaussian regression problem with common regressors reduces to a sequence of linear univariate Gaussian regression problems.

The linear univariate Gaussian regression problem is based on the stochastic model

$$\mathbf{y}_k = \theta' x_k + \mathbf{v}_k = x_k' \theta + \mathbf{v}_k \quad \mathbf{v}_k \sim N_{iid}(0, \sigma^2) \quad k = 1, 2, \dots, N \quad (\text{D.78})$$

in which  $\mathbf{y}_k : \Omega \mapsto \mathbb{R}$  and  $\mathbf{v}_k : \Omega \mapsto \mathbb{R}$  are stochastic variables, while the parameters,  $\theta \in \mathbb{R}^n$ , and the regressors,  $x_k \in \mathbb{R}^n$ , are deterministic variables. The noise sequence  $\{\mathbf{v}_k\}$  is assumed to arise from an independent identically normal distribution with zero mean and variance  $\sigma^2$ .

The model (D.78) may be expressed as the regression model (D.1)

$$\mathbf{y} = \Phi \theta + \mathbf{v} \quad \mathbf{v} \sim N(0, \sigma^2 I) \quad (\text{D.79})$$

in which the stochastic vectors are defined as

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_N \end{bmatrix} \quad \mathbf{v} = \begin{bmatrix} \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_N \end{bmatrix} \quad (\text{D.80})$$

and the regressors are given as

$$X = [x_1 \quad \dots \quad x_N] \quad (\text{D.81})$$

$$\Phi = X' = \begin{bmatrix} x_1' \\ \vdots \\ x_N' \end{bmatrix} \quad (\text{D.82})$$

The parameter vector is

$$\theta = \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_n \end{bmatrix} \quad (\text{D.83})$$

The covariance matrix of the noise is  $\Lambda = \sigma^2 I$ .

Alternatively, the regression model (D.78) may be expressed in the framework also used for multivariate regression (D.192)

$$\begin{aligned} \mathbf{Y} &= \theta' X + \mathbf{V} \\ &= AX + \mathbf{V} \end{aligned} \quad (\text{D.84})$$

in which

$$\mathbf{Y} = \mathbf{y}' = [\mathbf{y}_1 \quad \dots \quad \mathbf{y}_N] \quad (\text{D.85a})$$

$$\mathbf{V} = \mathbf{v}' = [\mathbf{v}_1 \quad \dots \quad \mathbf{v}_N] \quad (\text{D.85b})$$

and

$$A = \theta' = [\theta_1 \quad \dots \quad \theta_n] \quad (\text{D.86})$$

In the following subsections estimates of the parameters,  $\theta$ , will be based on a least squares criteria, the minimum variance criteria, and the maximum likelihood criterion. It turns out that all criteria leads to the least squares estimate, i.e.

$$\hat{\theta} = (\Phi' \Phi)^{-1} \Phi' y \quad (\text{D.87})$$

### D.2.1 Least Squares Estimate

The least squares estimate of the univariate regression model (D.78) is

$$\begin{aligned} \hat{\theta} &= \arg \min_{\theta} \left\{ \phi = \|y - \Phi \theta\|_2^2 \right\} \\ &= (\Phi' \Phi)^{-1} \Phi' y \end{aligned} \quad (\text{D.88})$$

This result is a direct consequence of corollary D.1.2.

### D.2.2 Weighted Least Squares Estimate

The weighted least squares estimate with weight  $W$  of the univariate linear regression model is obtained using proposition D.1.1

$$\begin{aligned} \hat{\theta} &= \arg \min_{\theta} \left\{ \phi = \|y - \Phi \theta\|_W^2 \right\} \\ &= (\Phi' W \Phi)^{-1} \Phi' W y \end{aligned} \quad (\text{D.89})$$

The minimum variance estimate is obtained with the weight matrix equal to the inverse of the covariance matrix:  $W = (\sigma^2 I)^{-1} = \sigma^{-2} I$ . With this weight matrix, the minimum variance estimate becomes

$$\begin{aligned} \hat{\theta} &= \arg \min_{\theta} \left\{ \phi = \|y - \Phi \theta\|_{W=\sigma^{-2} I}^2 \right\} \\ &= (\Phi' \sigma^{-2} I \Phi)^{-1} \Phi' \sigma^{-2} I y \\ &= (\Phi' \Phi)^{-1} \Phi' y \end{aligned} \quad (\text{D.90})$$

As is readily observed, the minimum variance estimate is equal to the least squares estimate in the univariate case.

### D.2.3 Maximum Likelihood Estimate

The maximum likelihood estimate for the univariate regression model (D.78) is developed for the case in which the noise covariance is known, and the case in which the noise covariance is unknown.

Using that the covariance is  $\Lambda = \sigma^2 I$ , the likelihood function of the univariate regression model (D.78) is

$$\begin{aligned} L(\theta, \sigma^2; y) &= L(\theta, \Lambda(\sigma^2); y) = -\ln p(y|\theta, \Lambda(\sigma^2)) \\ &= \frac{N}{2} \ln(2\pi) + \frac{1}{2} \ln(\det \Lambda) + \frac{1}{2} (y - \Phi\theta)' \Lambda^{-1} (y - \Phi\theta) \\ &= \frac{N}{2} \ln(2\pi) + \frac{N}{2} \ln(\sigma^2) + \frac{1}{2} \sigma^{-2} (y - \Phi\theta)' (y - \Phi\theta) \end{aligned} \quad (\text{D.91})$$

In the case of known noise covariance, the maximum likelihood estimate is

$$\hat{\theta} = \arg \min_{\theta} L(\theta, \sigma^2; y) \quad (\text{D.92})$$

In the case of unknown noise covariance, the maximum likelihood estimate is

$$(\hat{\theta}, \hat{\sigma}^2) = \arg \min_{\theta, \sigma^2} L(\theta, \sigma^2; y) \quad (\text{D.93})$$

#### D.2.3.1 Known Covariance

The maximum likelihood estimate of the univariate regression model (D.78) in the case of known noise covariance is stated in the next proposition.

**Proposition D.2.1 (ML estimate with known covariance)**

The maximum likelihood estimate

$$\hat{\theta} = \arg \min_{\theta} L(\theta, \sigma^2; y) \quad (\text{D.94})$$

of the univariate regression model (D.78) with known noise covariance,  $\sigma^2 I$ , is

$$\hat{\theta} = (\Phi' \Phi)^{-1} \Phi' y \quad (\text{D.95})$$

*Proof.* The likelihood function (D.91) of the univariate regression model is

$$\begin{aligned} L(\theta, \sigma^2; y) &= \frac{N}{2} \ln(2\pi\sigma^2) + \sigma^{-2} \frac{1}{2} (y - \Phi\theta)' (y - \Phi\theta) \\ &= \frac{N}{2} \ln(2\pi\sigma^2) + \sigma^{-2} \frac{1}{2} \|y - \Phi\theta\|_2^2 \end{aligned} \quad (\text{D.96})$$

Hence, the maximum likelihood estimate with known noise covariance,  $\sigma^2$ , is

$$\begin{aligned} \hat{\theta} &= \arg \min_{\theta} \left\{ L(\theta, \sigma^2; y) = \frac{N}{2} \ln(2\pi\sigma^2) + \sigma^{-2} \frac{1}{2} \|y - \Phi\theta\|_2^2 \right\} \\ &= \arg \min_{\theta} \frac{1}{2} \|y - \Phi\theta\|_2^2 \\ &= (\Phi' \Phi)^{-1} \Phi' y \end{aligned} \quad (\text{D.97})$$



□

The univariate maximum likelihood estimate with known covariance is equal to the least squares estimate. This is evident from the proof as well as the expression for the univariate maximum likelihood estimate.

### D.2.3.2 Unknown Covariance

The maximum likelihood estimate of the univariate regression model (D.78) is stated in the next proposition.

**Proposition D.2.2 (ML estimate with unknown covariance)**

The maximum likelihood estimate

$$(\hat{\theta}, \hat{\sigma}^2) = \arg \min_{\theta, \sigma^2} L(\theta, \sigma^2; y) \quad (\text{D.98})$$

of the univariate linear regression model (D.78) with unknown covariance,  $\sigma^2 I$ , is

$$\hat{\theta} = (\Phi' \Phi)^{-1} \Phi' y \quad (\text{D.99a})$$

$$\hat{\sigma}^2 = \frac{1}{N} \|y - \Phi \hat{\theta}\|_2^2 = \frac{1}{N} (y - \Phi \hat{\theta})' (y - \Phi \hat{\theta}) \quad (\text{D.99b})$$

*Proof.* The likelihood function (D.91) of the univariate regression model (D.78) is

$$\begin{aligned} L(\theta, \sigma^2; y) &= \frac{N}{2} \ln(2\pi) + \frac{N}{2} \ln(\sigma^2) + \frac{1}{2} \sigma^{-2} (y - \Phi \theta)' (y - \Phi \theta) \\ &= \frac{N}{2} \ln(2\pi) + \frac{N}{2} \ln(\sigma^2) + \frac{1}{2} \sigma^{-2} [y' y - 2(\Phi' y)' \theta + \theta' \Phi' \Phi \theta] \end{aligned} \quad (\text{D.100})$$

The first order necessary and sufficient conditions for the maximum likelihood estimate

$$(\hat{\theta}, \hat{\sigma}^2) = \arg \min_{\theta, \sigma^2} L(\theta, \sigma^2; y) \quad (\text{D.101})$$

are

$$\nabla_{\theta} L(\theta, \sigma^2; y) = \hat{\sigma}^{-2} [\Phi' \Phi \hat{\theta} - \Phi' y] = 0 \quad (\text{D.102a})$$

$$\nabla_{\sigma^2} L(\theta, \sigma^2; y) = \frac{N}{2} \hat{\sigma}^{-2} - \frac{1}{2} (\hat{\sigma}^2)^{-2} (y - \Phi \hat{\theta})' (y - \Phi \hat{\theta}) = 0 \quad (\text{D.102b})$$

(D.102a) yields the maximum likelihood estimate for the parameters,  $\theta$ :

$$\hat{\theta} = (\Phi' \Phi)^{-1} \Phi' y \quad (\text{D.103})$$

(D.102b) yields the maximum likelihood estimate for the noise variance,  $\sigma^2$ :

$$\hat{\sigma}^2 = \frac{1}{N} (y - \Phi \hat{\theta})' (y - \Phi \hat{\theta}) = \frac{1}{N} \|y - \Phi \hat{\theta}\|_2^2 \quad (\text{D.104})$$

□

### D.2.4 Inference

In assessing the properties of the least squares estimate, it is assumed that the data  $y$  are generated by the system  $\mathcal{S}$

$$\mathcal{S}: \quad \mathbf{y} = \Phi\theta + \mathbf{v} \quad \mathbf{v} \sim N(0, \sigma^2 I) \quad (\text{D.105})$$

in which  $\Phi \in \mathbb{R}^{N \times n}$  is a deterministic matrix. The stochastic vector,  $\mathbf{v}$ , is assumed to be normally distributed with zero mean and covariance  $\sigma^2 I$ , i.e.  $\mathbf{v} \sim N(0, \sigma^2 I)$ .  $\theta \in \mathbb{R}^n$  is the true parameter vector.

The least squares estimate,  $\hat{\theta} \in \mathbb{R}^n$ , of the parameter vector,  $\theta \in \mathbb{R}^n$ , given the data  $y \in \mathbb{R}^N$  is

$$\hat{\theta} = (\Phi' \Phi)^{-1} \Phi' y \quad (\text{D.106})$$

#### D.2.4.1 Distribution of the Least Squares Value Function

The expression for the least squares estimate

$$\hat{\theta} = \arg \min_{\theta} \left\{ V_N(\theta) = \frac{1}{2} \|y - \Phi\theta\|_2^2 \right\} = (\Phi' \Phi)^{-1} \Phi' y \quad (\text{D.107})$$

implies that the value function of the least squares estimate may be expressed as

$$\begin{aligned} V_N(\hat{\theta}) &= \min_{\theta} \left\{ V_N(\theta) = \frac{1}{2} \|y - \Phi\theta\|_2^2 \right\} \\ &= \frac{1}{2} \|y - \Phi(\Phi' \Phi)^{-1} \Phi' y\|_2^2 = \frac{1}{2} \|(I - \Phi(\Phi' \Phi)^{-1} \Phi') y\|_2^2 \\ &= \frac{1}{2} y' (I - \Phi(\Phi' \Phi)^{-1} \Phi')' (I - \Phi(\Phi' \Phi)^{-1} \Phi') y \\ &= \frac{1}{2} y' (I - \Phi(\Phi' \Phi)^{-1} \Phi') y \end{aligned} \quad (\text{D.108})$$

The value function  $V_N(\hat{\theta})$  is a realization of a stochastic function,  $\mathbf{V}_N$ , as  $y$  is a realization of a stochastic variable,  $\mathbf{y} = \Phi\theta + \mathbf{v}$ . The stochastic function  $\mathbf{V}_N(\hat{\theta})$  is given by

$$\begin{aligned} \mathbf{V}_N(\hat{\theta}) &= \frac{1}{2} \mathbf{y}' (I - \Phi(\Phi' \Phi)^{-1} \Phi') \mathbf{y} \\ &= \frac{1}{2} (\Phi\theta + \mathbf{v})' (I - \Phi(\Phi' \Phi)^{-1} \Phi') (\Phi\theta + \mathbf{v}) \\ &= \frac{1}{2} \mathbf{v}' (I - \Phi(\Phi' \Phi)^{-1} \Phi') \mathbf{v} \end{aligned} \quad (\text{D.109})$$

#### Lemma D.2.3

Let  $\Phi \in \mathbb{R}^{N \times n}$  have full column rank. Let the  $QR$  decomposition of  $\Phi$  be

$$\Phi = Q \begin{bmatrix} R \\ 0 \end{bmatrix} = [Q_1 \quad Q_2] \begin{bmatrix} R \\ 0 \end{bmatrix} = Q_1 R \quad (\text{D.110})$$

Then

$$I - \Phi(\Phi'\Phi)^{-1}\Phi' = Q_2Q_2' \quad (\text{D.111})$$

*Proof.*  $Q = [Q_1 \ Q_2]$  is the orthogonal matrix in a QR-decomposition of the full column rank matrix  $\Phi \in \mathbb{R}^{N \times n}$ . This implies  $Q_1 \in \mathbb{R}^{N \times n}$ ,  $Q_2 \in \mathbb{R}^{N \times (N-n)}$ ,  $Q_1'Q_1 = I_{n,n}$ ,  $Q_2'Q_2 = I_{N-n, N-n}$ , and  $QQ' = Q_1Q_1' + Q_2Q_2' = I_{N,N}$ .

Consequently

$$\Phi(\Phi'\Phi)^{-1}\Phi' = Q_1R(R'Q_1'Q_1R)^{-1}R'Q_1' = Q_1Q_1' \quad (\text{D.112})$$

and

$$I - \Phi(\Phi'\Phi)^{-1}\Phi' = I - Q_1Q_1' = Q_2Q_2' \quad (\text{D.113})$$

□

#### Lemma D.2.4

Let  $\mathbf{v} \sim N(0, \sigma^2 I)$  and define  $\mathbf{z}$  by

$$\mathbf{z} = \frac{1}{\sigma} Q_2' \mathbf{v} \quad (\text{D.114})$$

in which  $Q_2 \in \mathbb{R}^{N \times (N-n)}$  is an orthogonal matrix. Then the stochastic vector is distributed as

$$\mathbf{z} \sim N(0, I) \quad (\text{D.115})$$

and each component,  $\mathbf{z}_i$  for  $i = 1, \dots, N - n$ , is distributed as

$$\mathbf{z}_i \sim N(0, 1) \quad (\text{D.116})$$

*Proof.*  $\mathbf{z} = \frac{1}{\sigma} Q_2' \mathbf{v}$  is normally distributed as it is a linear combination of a normally distributed vector,  $\mathbf{v}$ . The mean is  $E\{\mathbf{z}\} = E\left\{\frac{1}{\sigma} Q_2' \mathbf{v}\right\} = \frac{1}{\sigma} Q_2' E\{\mathbf{v}\} = 0$  and the variance is

$$\begin{aligned} V\{\mathbf{z}\} &= \langle \mathbf{z}, \mathbf{z} \rangle = \left\langle \frac{1}{\sigma} Q_2' \mathbf{v}, \frac{1}{\sigma} Q_2' \mathbf{v} \right\rangle = \frac{1}{\sigma} Q_2' \langle \mathbf{v}, \mathbf{v} \rangle Q_2 \frac{1}{\sigma} = \frac{1}{\sigma} Q_2' \sigma^2 I Q_2 \frac{1}{\sigma} \\ &= Q_2' Q_2 = I_{N-n, N-n} = I \end{aligned} \quad (\text{D.117})$$

Hence  $\mathbf{z} \sim N(0, I)$ .  $\mathbf{z}_i \sim N(0, 1)$  for  $i = 1, 2, \dots, N - n$  follows trivially. □

#### Proposition D.2.5

Let  $\mathbf{V}_N(\hat{\theta})$  be the least squares value function. Let the regression matrix  $\Phi \in \mathbb{R}^{N \times n}$  have full column rank. Let  $\mathbf{v} \sim N(0, \sigma^2 I)$ .

Then

$$\frac{2\mathbf{V}_N(\hat{\theta})}{\sigma^2} \sim \chi^2(N - n) \quad (\text{D.118})$$

*Proof.* It follows from (D.108), (D.109), and lemma D.2.3 that the stochastic value function can be expressed as

$$\mathbf{V}_N(\hat{\theta}) = \frac{1}{2} \mathbf{v}' (I - \Phi(\Phi'\Phi)^{-1}\Phi') \mathbf{v} = \frac{1}{2} \mathbf{v}' Q_2 Q_2' \mathbf{v} = \frac{1}{2} (Q_2' \mathbf{v})' (Q_2' \mathbf{v}) \quad (\text{D.119})$$

in which  $Q_2 \in \mathbb{R}^{N \times (N-n)}$  is an orthogonal matrix in the QR decomposition of  $\Phi \in \mathbb{R}^{N \times n}$ , i.e.  $\Phi = Q \begin{bmatrix} R \\ 0 \end{bmatrix} = [Q_1 \quad Q_2] \begin{bmatrix} R \\ 0 \end{bmatrix} = Q_1 R$ . Define  $\mathbf{z} = \frac{1}{\sigma} Q_2' \mathbf{v}$  and use lemma D.2.4 as well as the definition of the  $\chi^2$ -distribution to establish

$$\begin{aligned} \frac{2\mathbf{V}_N(\hat{\theta})}{\sigma^2} &= \frac{1}{\sigma^2} (Q_2' \mathbf{v})' (Q_2' \mathbf{v}) = \left( \frac{1}{\sigma} Q_2' \mathbf{v} \right)' \left( \frac{1}{\sigma} Q_2' \mathbf{v} \right) \\ &= \mathbf{z}' \mathbf{z} = \sum_{i=1}^{N-n} \mathbf{z}_i^2 \sim \chi^2(N-n) \end{aligned} \quad (\text{D.120})$$

□

#### D.2.4.2 Unbiased Estimate of the Noise Variance

The maximum likelihood estimate,  $\hat{\sigma}^2$ , of the noise variance,  $\sigma^2$ , is

$$\hat{\sigma}^2 = \frac{1}{N} \left\| y - \Phi \hat{\theta} \right\|_2^2 = \frac{2}{N} V_N(\hat{\theta}) \quad (\text{D.121})$$

By proposition D.2.5 it follows that

$$N \frac{\hat{\sigma}^2}{\sigma^2} = \frac{2\mathbf{V}_N(\hat{\theta})}{\sigma^2} \sim \chi^2(N-n) \quad (\text{D.122})$$

The distribution  $\chi^2(n)$  has the mean  $n$  and variance  $2n$ . Hence

$$E \left\{ N \frac{\hat{\sigma}^2}{\sigma^2} \right\} = \frac{N}{\sigma^2} E \{ \hat{\sigma}^2 \} = N - n \quad (\text{D.123})$$

which implies that the maximum likelihood estimate,  $\hat{\sigma}^2$ , of the noise variance,  $\sigma^2$ , is biased

$$E \{ \hat{\sigma}^2 \} = \frac{N-n}{N} \sigma^2 = \left( 1 - \frac{n}{N} \right) \sigma^2 \quad (\text{D.124})$$

#### Proposition D.2.6 (Unbiased Estimate of the Noise Variance)

Define

$$\hat{s}^2 = \frac{1}{N-n} \left\| y - \Phi \hat{\theta} \right\|_2^2 = \frac{2}{N-n} V_N(\hat{\theta}) \quad (\text{D.125})$$

Then  $\hat{s}^2$  has the distribution given by

$$(N-n) \frac{\hat{s}^2}{\sigma^2} \sim \chi^2(N-n) \quad (\text{D.126})$$

and  $\hat{s}^2$  is an unbiased estimate of the noise variance,  $\sigma^2$ .

*Proof.* The definition of  $\hat{s}^2$

$$\hat{s}^2 = \frac{2}{N-n} \mathbf{V}_N(\hat{\theta}) \quad (\text{D.127})$$

implies

$$(N-n) \frac{\hat{s}^2}{\sigma^2} = \frac{2\mathbf{V}_N(\hat{\theta})}{\sigma^2} \sim \chi^2(N-n) \quad (\text{D.128})$$

and

$$E \left\{ (N-n) \frac{\hat{s}^2}{\sigma^2} \right\} = \frac{N-n}{\sigma^2} E \{ \hat{s}^2 \} = N-n \quad (\text{D.129})$$

Hence,  $E \{ \hat{s}^2 \} = \sigma^2$  and  $\hat{s}^2$  is an unbiased estimate of  $\sigma^2$ .  $\square$

It follows immediately that the unbiased noise variance estimate,  $\hat{s}^2$ , is related to the maximum likelihood noise variance estimate,  $\hat{\sigma}^2$ , by

$$\hat{s}^2 = \frac{2V_N(\hat{\theta})}{N-n} = \frac{N}{N-n} \hat{\sigma}^2 \quad (\text{D.130})$$

The statistical properties of the unbiased and maximum likelihood noise variance estimate can be summarized by

$$(N-n) \frac{\hat{s}^2}{\sigma^2} \sim \chi^2(N-n) \quad N \frac{\hat{\sigma}^2}{\sigma^2} \sim \chi^2(N-n) \quad (\text{D.131a})$$

$$E \{ \hat{s}^2 \} = \sigma^2 \quad E \{ \hat{\sigma}^2 \} = \left( 1 - \frac{n}{N} \right) \sigma^2 \quad (\text{D.131b})$$

$$V \{ \hat{s}^2 \} = 2 \frac{1}{N-n} \sigma^4 \quad V \{ \hat{\sigma}^2 \} = 2 \frac{N-n}{N^2} \sigma^4 \quad (\text{D.131c})$$

From this comparison, it is apparent that the maximum likelihood estimate,  $\hat{\sigma}^2$ , is biased but has a smaller variance than the unbiased estimate,  $\hat{s}^2$ :

$$\frac{V \{ \hat{\sigma}^2 \}}{V \{ \hat{s}^2 \}} = \left( \frac{N-n}{N} \right)^2 = \left( 1 - \frac{n}{N} \right)^2 < 1 \quad 0 < n < N \quad (\text{D.132})$$

This is a consequence of the general bias-variance dilemma (Hastie *et al.*, 2001).

#### D.2.4.3 Distribution of the Estimate

In the following the distribution of the least squares estimate is established. This result is used to deduce the confidence interval and marginal distribution of the least square parameters for the case when the covariance is computed as  $\hat{P} = \hat{s}^2(\Phi'\Phi)^{-1}$  and  $\hat{P} = \hat{\sigma}^2(\Phi'\Phi)^{-1}$ , respectively. It should be noted that the least squares estimate is identical to the maximum likelihood estimate for the univariate regression model.

**Proposition D.2.7 (Distribution of the Least Squares Estimate)**

Let the data,  $y$ , be generated by the system (D.105). Let  $\mathbf{v} \sim N(0, \sigma^2 I)$ . Then the least squares estimate is distributed as

$$\hat{\boldsymbol{\theta}} \sim N(\boldsymbol{\theta}, \sigma^2(\Phi' \Phi)^{-1}) \quad (\text{D.133})$$

*Proof.* The least squares estimate may be expressed as

$$\hat{\boldsymbol{\theta}} = (\Phi' \Phi)^{-1} \Phi' \mathbf{y} = (\Phi' \Phi)^{-1} \Phi' (\Phi \boldsymbol{\theta} + \mathbf{v}) = \boldsymbol{\theta} + (\Phi' \Phi)^{-1} \Phi' \mathbf{v} \quad (\text{D.134})$$

This implies that the estimate is normally distributed as it is related linearly to a normally distributed variable,  $\mathbf{v}$ . The mean is

$$\bar{\boldsymbol{\theta}} = E \{ \hat{\boldsymbol{\theta}} \} = \boldsymbol{\theta} + (\Phi' \Phi)^{-1} \Phi' E \{ \mathbf{v} \} = \boldsymbol{\theta} \quad (\text{D.135})$$

and the covariance is

$$\begin{aligned} E \{ (\hat{\boldsymbol{\theta}} - \bar{\boldsymbol{\theta}})(\hat{\boldsymbol{\theta}} - \bar{\boldsymbol{\theta}})' \} &= [(\Phi' \Phi)^{-1} \Phi'] E \{ \mathbf{v} \mathbf{v}' \} [(\Phi' \Phi)^{-1} \Phi']' \\ &= [(\Phi' \Phi)^{-1} \Phi'] [\sigma^2 I] [\Phi (\Phi' \Phi)^{-1}] = \sigma^2 (\Phi' \Phi)^{-1} \end{aligned} \quad (\text{D.136})$$

Consequently

$$\hat{\boldsymbol{\theta}} \sim N(\boldsymbol{\theta}, \sigma^2(\Phi' \Phi)^{-1}) \quad (\text{D.137})$$

□

**Proposition D.2.8 (Confidence interval, known covariance)**

Let

$$\hat{\boldsymbol{\theta}} \sim N(\boldsymbol{\theta}, P) \quad (\text{D.138})$$

in which  $\boldsymbol{\theta} \in \mathbb{R}^n$  and  $P = \sigma^2(\Phi' \Phi)^{-1}$  is a known deterministic real symmetric positive definite matrix,  $P \in \mathbb{R}^{n \times n}$ .  $\sigma$  is a known real number. Then

$$\hat{\theta}_i - \theta_i \sim N(0, P_{ii}) \quad (\text{D.139a})$$

$$\frac{\hat{\theta}_i - \theta_i}{\sqrt{P_{ii}}} \sim N(0, 1) \quad (\text{D.139b})$$

and

$$(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})' P^{-1} (\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}) \sim \chi^2(n) \quad (\text{D.140})$$

*Proof.* (D.139) follows trivially. (D.139a) comes from the definition of a marginal distribution. (D.139b) is obtained using the covariance properties associated with the normal distribution.

To derive (D.140) Cholesky factorize  $P$

$$P = LL' \quad (\text{D.141})$$

and define the stochastic variable  $\mathbf{z}$  as

$$\mathbf{z} = L^{-1} (\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}) \quad (\text{D.142})$$

Note that  $\mathbf{z} \sim N(0, I)$ .  $\mathbf{z}$  is normally distributed as it is a linear combination of a normally distributed variable.  $E\{\mathbf{z}\} = L^{-1} \left( E\{\hat{\boldsymbol{\theta}}\} - \boldsymbol{\theta} \right) = 0$  and

$$\begin{aligned} \langle \mathbf{z}, \mathbf{z} \rangle &= \langle L^{-1} (\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}), L^{-1} (\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}) \rangle = L^{-1} \langle \hat{\boldsymbol{\theta}} - \boldsymbol{\theta}, \hat{\boldsymbol{\theta}} - \boldsymbol{\theta} \rangle (L^{-1})' \\ &= L^{-1} P (L^{-1})' = L^{-1} L L' (L')^{-1} = I \end{aligned} \quad (\text{D.143})$$

Consequently,  $\sum_{i=1}^n \mathbf{z}_i^2 \sim \chi^2(n)$  and

$$\begin{aligned} \sum_{i=1}^n \mathbf{z}_i^2 &= \mathbf{z}' \mathbf{z} = \left[ L^{-1} (\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}) \right]' L^{-1} (\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}) \\ &= (\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})' (L^{-1})' L^{-1} (\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}) = (\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})' (L L')^{-1} (\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}) \\ &= (\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})' P^{-1} (\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}) \end{aligned} \quad (\text{D.144})$$

which proves (D.140).  $\square$

Without proof we use the well-known result for Hilbert spaces, that the least squares estimate,  $\hat{\boldsymbol{\theta}}$ , and the residuals,  $\mathbf{e} = \mathbf{y} - \Phi \hat{\boldsymbol{\theta}}$ , are orthogonal. For normally distributed variables this implies that they are independent. Consequently,  $\hat{\boldsymbol{\theta}}$  and  $\mathbf{V} = \frac{1}{2} \mathbf{e}' \mathbf{e}$  are independent. Further this implies that  $\hat{\boldsymbol{\theta}}$  and  $\hat{\sigma}^2$  are independent as well as independence of  $\hat{\boldsymbol{\theta}}$  and  $\hat{\sigma}^2$ . These independence properties are used to establish the following results concerning the distribution of the least squares estimate in the case of unknown covariance of  $\mathbf{v}$  in the univariate regression model.

**Proposition D.2.9 (Confidence interval, unknown covariance)**

Let

$$\hat{\boldsymbol{\theta}} \sim N(\boldsymbol{\theta}, P) \quad (\text{D.145})$$

in which  $\boldsymbol{\theta} \in \mathbb{R}^n$ ,  $\hat{P} = \sigma^2 (\Phi' \Phi)^{-1} \in \mathbb{R}^{n \times n}$  is a positive definite matrix. Define  $\hat{P} = \hat{s}^2 (\Phi' \Phi)^{-1}$  and  $\hat{s}^2 = \frac{1}{N-n} \left\| \mathbf{y} - \Phi \hat{\boldsymbol{\theta}} \right\|_2^2$ . Note that  $\hat{s}^2$  stems from the distribution

$$(N-n) \frac{\hat{s}^2}{\sigma^2} \sim \chi^2(N-n) \quad (\text{D.146})$$

Further define  $\hat{\mathbf{P}} = \hat{s}^2 (\Phi' \Phi)^{-1}$ . Assume that  $\hat{\boldsymbol{\theta}}$  and  $\hat{s}^2$  are independent stochastic variables.

Then, the marginal distribution of  $\hat{\boldsymbol{\theta}}_i$  is

$$\frac{\hat{\boldsymbol{\theta}}_i - \theta_i}{\sqrt{\hat{P}_{ii}}} \sim t(N-n) \quad (\text{D.147})$$

and the confidence interval of  $\hat{\boldsymbol{\theta}}$  may be computed from the relation

$$\frac{1}{n} (\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})' \hat{\mathbf{P}}^{-1} (\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}) \sim F(n, N-n) \quad (\text{D.148})$$

*Proof.* Let  $Q_{ii} = [(\Phi'\Phi)^{-1}]_{ii}$ ,  $P_{ii} = \sigma^2 [(\Phi'\Phi)^{-1}]_{ii} = \sigma^2 Q_{ii}$ ,  $\hat{P}_{ii} = \hat{s}^2 [(\Phi'\Phi)^{-1}]_{ii} = \hat{s}^2 Q_{ii}$ , and  $\hat{P}_{ii} = \hat{s}^2 [(\Phi'\Phi)^{-1}]_{ii} = \hat{s}^2 Q_{ii}$ . Note

$$\frac{\hat{\theta}_i - \theta_i}{\sqrt{P_{ii}}} \sim N(0, 1) \quad (\text{D.149})$$

Consequently

$$t = \frac{\hat{\theta}_i - \theta_i}{\sqrt{P_{ii}}} \frac{\sqrt{N-n}}{\sqrt{(N-n)\frac{\hat{s}^2}{\sigma^2}}} \sim t(N-n) \quad (\text{D.150})$$

and

$$\begin{aligned} t &= \frac{\hat{\theta}_i - \theta_i}{\sqrt{P_{ii}}} \frac{\sqrt{N-n}}{\sqrt{(N-n)\frac{\hat{s}^2}{\sigma^2}}} = \frac{\hat{\theta}_i - \theta_i}{\sqrt{\sigma^2 Q_{ii}}} \frac{\sqrt{N-n}}{\sqrt{(N-n)\frac{\hat{s}^2}{\sigma^2}}} = \frac{\hat{\theta}_i - \theta_i}{\sqrt{\hat{s}^2 Q_{ii}}} \\ &= \frac{\hat{\theta}_i - \theta_i}{\sqrt{\hat{P}_{ii}}} \sim t(N-n) \end{aligned} \quad (\text{D.151})$$

Note that  $(\hat{\theta} - \theta)' P^{-1} (\hat{\theta} - \theta) \sim \chi^2(n)$ . Then

$$f = \frac{(\hat{\theta} - \theta)' P^{-1} (\hat{\theta} - \theta)}{n} \frac{N-n}{(N-n)\frac{\hat{s}^2}{\sigma^2}} \sim F(n, N-n) \quad (\text{D.152})$$

which implies

$$\begin{aligned} f &= \frac{(\hat{\theta} - \theta)' P^{-1} (\hat{\theta} - \theta)}{n} \frac{N-n}{(N-n)\frac{\hat{s}^2}{\sigma^2}} \\ &= \frac{(\hat{\theta} - \theta)' (\Phi'\Phi) (\hat{\theta} - \theta)}{n\sigma^2} \frac{N-n}{(N-n)\frac{\hat{s}^2}{\sigma^2}} \\ &= \frac{(\hat{\theta} - \theta)' [\hat{s}^2 (\Phi'\Phi)^{-1}]^{-1} (\hat{\theta} - \theta)}{n} \\ &= \frac{(\hat{\theta} - \theta)' \hat{P}^{-1} (\hat{\theta} - \theta)}{n} \sim F(n, N-n) \end{aligned} \quad (\text{D.153})$$

□

**Proposition D.2.10 (Confidence interval, unknown covariance)**

Let

$$\hat{\theta} \sim N(\theta, P) \quad (\text{D.154})$$

in which  $\theta \in \mathbb{R}^n$ ,  $P = \sigma^2(\Phi'\Phi)^{-1} \in \mathbb{R}^{n \times n}$  is a positive definite matrix. Define  $\hat{P} = \hat{\sigma}^2(\Phi'\Phi)^{-1}$  and  $\hat{\sigma}^2 = \frac{1}{N} \|y - \Phi\hat{\theta}\|_2^2$ . Note that  $\hat{\sigma}^2$  stems from the distribution

$$N \frac{\hat{\sigma}^2}{\sigma^2} \sim \chi^2(N-n) \quad (\text{D.155})$$



Further define  $\hat{\mathbf{P}} = \hat{\sigma}^2(\Phi'\Phi)^{-1}$ . Assume that  $\hat{\boldsymbol{\theta}}$  and  $\hat{\sigma}^2$  are independent stochastic variables.

Then the marginal distribution of  $\hat{\theta}_i$  is

$$\sqrt{\frac{N-n}{N}} \frac{\hat{\theta}_i - \theta_i}{\sqrt{\hat{P}_{ii}}} \sim t(N-n) \quad (\text{D.156})$$

and the confidence interval of  $\hat{\boldsymbol{\theta}}$  may be computed from the relation

$$\frac{N-n}{N} \frac{(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})' \hat{\mathbf{P}}^{-1} (\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})}{n} \sim F(n, N-n) \quad (\text{D.157})$$

*Proof.* Let  $Q_{ii} = [(\Phi'\Phi)^{-1}]_{ii}$ ,  $P_{ii} = \sigma^2 [(\Phi'\Phi)^{-1}]_{ii} = \sigma^2 Q_{ii}$ ,  $\hat{P}_{ii} = \hat{\sigma}^2 [(\Phi'\Phi)^{-1}]_{ii} = \hat{\sigma}^2 Q_{ii}$ , and  $\hat{\mathbf{P}}_{ii} = \hat{\sigma}^2 [(\Phi'\Phi)^{-1}]_{ii} = \hat{\sigma}^2 Q_{ii}$ . Note

$$\frac{\hat{\theta}_i - \theta_i}{\sqrt{\hat{P}_{ii}}} \sim N(0, 1) \quad (\text{D.158})$$

Consequently

$$t = \frac{\hat{\theta}_i - \theta_i}{\sqrt{\hat{P}_{ii}}} \frac{\sqrt{N-n}}{\sqrt{N \frac{\hat{\sigma}^2}{\sigma^2}}} \sim t(N-n) \quad (\text{D.159})$$

and

$$\begin{aligned} t &= \frac{\hat{\theta}_i - \theta_i}{\sqrt{\hat{P}_{ii}}} \frac{\sqrt{N-n}}{\sqrt{N \frac{\hat{\sigma}^2}{\sigma^2}}} = \frac{\hat{\theta}_i - \theta_i}{\sqrt{\sigma^2 Q_{ii}}} \frac{\sqrt{N-n}}{\sqrt{N \frac{\hat{\sigma}^2}{\sigma^2}}} = \sqrt{\frac{N-n}{N}} \frac{\hat{\theta}_i - \theta_i}{\sqrt{\hat{\sigma}^2 Q_{ii}}} \\ &= \sqrt{\frac{N-n}{N}} \frac{\hat{\theta}_i - \theta_i}{\sqrt{\hat{\mathbf{P}}_{ii}}} \sim t(N-n) \end{aligned} \quad (\text{D.160})$$

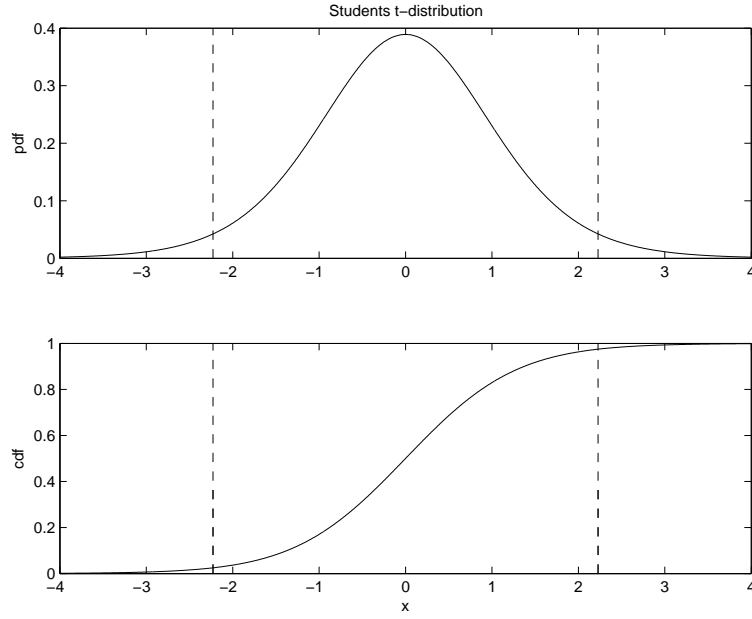
Note that  $(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})' \mathbf{P}^{-1} (\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}) \sim \chi^2(n)$ . Then

$$\mathbf{f} = \frac{(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})' \mathbf{P}^{-1} (\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})}{n} \frac{N-n}{N \frac{\hat{\sigma}^2}{\sigma^2}} \sim F(n, N-n) \quad (\text{D.161})$$

which implies

$$\begin{aligned} \mathbf{f} &= \frac{(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})' \mathbf{P}^{-1} (\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})}{n} \frac{N-n}{N \frac{\hat{\sigma}^2}{\sigma^2}} \\ &= \frac{(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})' (\Phi'\Phi) (\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})}{n \sigma^2} \frac{N-n}{N \frac{\hat{\sigma}^2}{\sigma^2}} \\ &= \frac{N-n}{N} \frac{(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})' [\hat{\sigma}^2 (\Phi'\Phi)^{-1}]^{-1} (\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})}{n} \\ &= \frac{N-n}{N} \frac{(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})' \hat{\mathbf{P}}^{-1} (\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})}{n} \sim F(n, N-n) \end{aligned} \quad (\text{D.162})$$

□



**Figure D.1.** Probability density function (top) and cumulative probability density function (bottom) of Student's t-distribution with 10 degrees of freedom. The dotted lines indicate the 95% confidence interval  $[-2.2281, 2.2281]$ .

Let  $p = t(x; V)$  denote the cumulative Student's t-distribution with  $V$  degrees of freedom evaluated at  $x$ . Let  $x = t^{-1}(p; V)$  denote the corresponding inverse function. Then the  $(1 - \alpha)$  marginal confidence interval for the estimated parameters may be expressed as

$$\left[ \hat{\theta}_i + t^{-1}\left(\frac{\alpha}{2}; N - n\right) \sqrt{\hat{P}_{ii}} \quad \hat{\theta}_i + t^{-1}\left(1 - \frac{\alpha}{2}; N - n\right) \sqrt{\hat{P}_{ii}} \right] \quad (\text{D.163})$$

in which the covariance matrix  $\hat{P}$  is computed as

$$\hat{P} = \hat{s}^2 (\Phi' \Phi)^{-1} \quad (\text{D.164})$$

using  $\hat{s}^2 = \frac{(y - \Phi \hat{\theta})'(y - \Phi \hat{\theta})}{N - n}$  and  $\hat{\theta} = (\Phi' \Phi)^{-1} \Phi' y$ .

#### D.2.4.4 Model Discrimination

The following results are useful for discriminating between two univariate linear regression models,  $\mathcal{M}_1$  and  $\mathcal{M}_2$  (Söderström and Stoica, 1989; Madsen, 1995; Poulsen, 1995a; Ljung, 1999).

**Lemma D.2.11**

Let the data  $\{y_t\}_{t=1}^N$  be generated by the univariate regression model (D.78), i.e.

$$\mathcal{S} : \quad \mathbf{y}_t = x_t' \theta + \mathbf{v}_t \quad \mathbf{v}_t \sim N_{iid}(0, \sigma^2) \quad t = 1, 2, \dots, N \quad (\text{D.165})$$

Let the regressors of model  $\mathcal{M}_1$  be identical to the regressors of the system,  $\mathcal{S}$

$$\mathcal{M}_1 : \quad \hat{\theta}_1 = \arg \min_{\theta} \left\{ V_1(\theta) = \frac{1}{2} \sum_{t=1}^N \|y_t - x_t' \theta\|_2^2 \right\} \quad (\text{D.166})$$

Let model,  $\mathcal{M}_2$ , be a superset of model  $\mathcal{M}_1$ , i.e.  $\mathcal{M}_2 \supset \mathcal{M}_1$ . Denote the corresponding parameter estimates as

$$\mathcal{M}_2 : \quad (\hat{\theta}_2, \hat{\eta}) = \arg \min_{\theta, \eta} \left\{ V_2(\theta, \eta) = \frac{1}{2} \sum_{t=1}^N \|y_t - x_t' \theta - z_t' \eta\|_2^2 \right\} \quad (\text{D.167})$$

Let  $n_1 = \dim(\theta) = \dim(x_t)$  and  $n_2 = \dim(\theta) + \dim(\eta) = \dim(x_t) + \dim(z_t)$ .

Then

1. The value function  $V_2(\hat{\theta}_2, \hat{\eta})$  is a realization of the stochastic variable  $V_2(\hat{\theta}_2, \hat{\eta})$  distributed as

$$\frac{2V_2(\hat{\theta}_2, \hat{\eta})}{\sigma^2} \sim \chi^2(N - n_2) \quad (\text{D.168})$$

2. The difference,  $V_1(\hat{\theta}_1) - V_2(\hat{\theta}_2, \hat{\eta})$ , is a realization of the stochastic variable  $V_1(\hat{\theta}_1) - V_2(\hat{\theta}_2, \hat{\eta})$  distributed as

$$\frac{2(V_1(\hat{\theta}_1) - V_2(\hat{\theta}_2, \hat{\eta}))}{\sigma^2} \sim \chi^2(n_2 - n_1) \quad (\text{D.169})$$

3.  $V_1(\hat{\theta}_1) - V_2(\hat{\theta}_2, \hat{\eta})$  and  $V_2(\hat{\theta}_2, \hat{\eta})$  are independent random variables.

*Proof.* See Söderström and Stoica (1989).  $\square$

**Proposition D.2.12**

Let the assumption of lemma D.2.11 be satisfied. Then

$$\mathbf{f} = \frac{V_1(\hat{\theta}_1) - V_2(\hat{\theta}_2, \hat{\eta})}{n_2 - n_1} \frac{N - n_2}{V_2(\hat{\theta}_2, \hat{\eta})} \sim F(n_2 - n_1, N - n_2) \quad (\text{D.170})$$

*Proof.* The result follows trivially from lemma D.2.11. See also Söderström and Stoica (1989).  $\square$

**Remark D.2.13**

Proposition D.2.12 may be used to construct a statistical test for comparison of the model structures  $\mathcal{M}_1$  and  $\mathcal{M}_2$ . Let  $\hat{\theta}_1$  be parameters in the model  $\mathcal{M}_1$ . Estimate parameters  $(\hat{\theta}_2, \hat{\eta})$  in a model  $\mathcal{M}_2 \supset \mathcal{M}_1$ . If  $f$  computed by (D.170) is in the  $(1 - \alpha)$ -confidence interval of  $F(n_2 - n_1, N - n_2)$  then model  $\mathcal{M}_2$  does not produce a loss function with a statistical significant lower value; hence, we cannot reject model  $\mathcal{M}_1$ . Consequently, either we accept model  $\mathcal{M}_1$  or conduct other tests to see if these reject model  $\mathcal{M}_1$ . If  $f$  is not in the  $(1 - \alpha)$ -confidence interval, we reject model  $\mathcal{M}_1$  because model  $\mathcal{M}_2$  gives a loss function that has a lower value that is statistically significant.

**Remark D.2.14**

Asymptotically

$$f = \frac{V_1(\hat{\theta}_1) - V_2(\hat{\theta}_2, \hat{\eta})}{n_2 - n_1} \frac{N - n_2}{V_2(\hat{\theta}_2, \hat{\eta})} \sim F(n_2 - n_1, N - n_2) \rightarrow \chi^2(n_2 - n_1) \quad N \rightarrow \infty \quad (\text{D.171})$$

Proposition D.2.12 is typically applied by computing the least squares estimate and associated value function (loss function) for a sequence of successively larger regressors. The value function is plotted as function of the number of parameters and proposition D.2.12 is used to select the minimum number of parameters for which no further statistically significant reduction in the value function is obtained (Söderström and Stoica, 1989; Madsen, 1995; Poulsen, 1995a).

**D.2.5 Maximum a Posteriori Estimation**

Maximum *a posteriori* estimation is also called Bayes estimation or Bayesian estimation (Hamilton, 1994).

Let  $\theta \in \mathbb{R}^p$  be a parameter to be estimated based on a sample observations,  $y$ , stemming from some distribution. Classical statistics, i.e. maximum likelihood based estimation, assumes that there exists some true value of  $\theta$ . This true value is regarded as an unknown but fixed number. An estimator  $\hat{\theta}$  is constructed from the data, and  $\hat{\theta}$  is therefore a random variable. The efficiency of the estimator is judged by the mean squared error of the random variable, i.e.  $E\{(\hat{\theta} - \theta)(\hat{\theta} - \theta)\}$ .

In Bayesian statistics, by contrast,  $\theta$  itself is regarded as a stochastic variable. All inference about  $\theta$  takes the form of statements of probability. The view is that there will always exist some uncertainty about  $\theta$ , and the goal of statistical analysis is to describe this uncertainty in terms of a probability distribution. Any information about  $\theta$  available prior to the observation process is represented by a *prior* distribution, i.e. a prior probability density function  $p(\theta)$ . The joint likelihood of  $y$  and  $\theta$  is expressed as

$$p(y, \theta) = p(\theta|y)p(y) = p(y|\theta)p(\theta) \quad (\text{D.172})$$

and used to derive Bayes law for the *posterior* probability density of  $\theta$

$$p(\theta|y) = \frac{p(y, \theta)}{p(y)} = \frac{p(y|\theta)p(\theta)}{p(y)} \propto p(y|\theta)p(\theta) \quad (\text{D.173})$$

By Bayes law it is evident that the *posterior* probability density,  $p(\theta|y)$ , is proportional to the likelihood,  $p(y|\theta)$ , and the *prior* probability density,  $p(\theta)$ .

In some simple cases, the probability density,  $p(\theta|y)$ , and thereby the distribution of the *a posteriori* variable  $\theta|y$  may be expressed analytically. However, in

most cases this is not possible; even for linear models. This can be due to the prior probability distribution, i.e. the prior probability density does not lead to an analytical expression for the posterior probability density. In these cases it is common to approximate the *posterior* mean

$$E\{\boldsymbol{\theta}|\mathbf{y}\} = \int \boldsymbol{\theta} p(\boldsymbol{\theta}|\mathbf{y}) d\boldsymbol{\theta} \quad (\text{D.174})$$

by the *posterior* mode

$$\hat{\boldsymbol{\theta}} = \arg \max_{\boldsymbol{\theta}} \{p(\boldsymbol{\theta}|\mathbf{y})\} = \arg \max_{\boldsymbol{\theta}} \{p(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})\} \quad (\text{D.175})$$

Consequently, usually the *a posteriori* estimate must be computed numerically as the mode. This topic is outside the scope of this treatment. In the following some examples are provided in which it is possible to characterize the posterior distribution analytically.

### D.2.5.1 Known Covariance

Consider the Bayesian univariate regression model

$$\mathbf{y} = \Phi\boldsymbol{\theta} + \mathbf{v} \quad \mathbf{v} \sim N(0, \sigma^2 I), \boldsymbol{\theta} \sim N(\boldsymbol{\theta}_0, P_{\boldsymbol{\theta}_0}) \quad (\text{D.176})$$

and note that in contrast to the univariate regression model (D.79), the parameter  $\boldsymbol{\theta}$  is a stochastic variable and not a fixed variable. The density distributions,  $p(\boldsymbol{\theta})$  and  $p(\mathbf{y}|\boldsymbol{\theta})$ , for the stochastic model (D.176) are

$$p(\boldsymbol{\theta}) = (2\pi)^{-p/2} (\det P_{\boldsymbol{\theta}_0})^{-1/2} \exp\left(-\frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}_0)' P_{\boldsymbol{\theta}_0}^{-1}(\boldsymbol{\theta} - \boldsymbol{\theta}_0)\right) \quad (\text{D.177})$$

and

$$\begin{aligned} p(\mathbf{y}|\boldsymbol{\theta}) &= p_v(\mathbf{y} - \Phi\boldsymbol{\theta}) \\ &= (2\pi)^{-n/2} (\det \sigma^2 I)^{-1/2} \exp\left(-\frac{1}{2}(\mathbf{y} - \Phi\boldsymbol{\theta})'(\sigma^2 I)^{-1}(\mathbf{y} - \Phi\boldsymbol{\theta})\right) \\ &= (2\pi\sigma^2)^{-n/2} \exp\left(-\frac{1}{2\sigma^2} \|\mathbf{y} - \Phi\boldsymbol{\theta}\|_2^2\right) \end{aligned} \quad (\text{D.178})$$

Consequently, the posterior mode maximization problem (D.175) may be stated as

$$\hat{\boldsymbol{\theta}} = \arg \max_{\boldsymbol{\theta}} \{p(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})\} = \arg \min_{\boldsymbol{\theta}} L(\boldsymbol{\theta}) \quad (\text{D.179})$$

in which

$$\begin{aligned} L(\boldsymbol{\theta}) &= -\ln[p(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})] = -\ln p(\mathbf{y}|\boldsymbol{\theta}) - \ln p(\boldsymbol{\theta}) \\ &= \left[ \frac{(n+p)}{2} \ln(2\pi) + \frac{n}{2} \ln(\sigma^2) + \frac{1}{2} \ln(\det P_{\boldsymbol{\theta}_0}) \right] \\ &\quad + \frac{1}{2} \|\boldsymbol{\theta} - \boldsymbol{\theta}_0\|_{P_{\boldsymbol{\theta}_0}^{-1}}^2 + \frac{1}{2\sigma^2} \|\mathbf{y} - \Phi\boldsymbol{\theta}\|_2^2 \end{aligned} \quad (\text{D.180})$$

The posterior estimate,  $\hat{\theta}$ , computed as the mode of  $p(\theta|y)$  is stated in the following proposition.

**Proposition D.2.15 (MAP estimate with known covariance)**

The maximum a posteriori estimate

$$\hat{\theta} = \arg \max_{\theta} p(\theta|y) = \arg \min_{\theta} L(\theta) \quad (\text{D.181})$$

of the univariate regression model (D.176) with known noise covariance,  $\sigma^2 I$ , is

$$\hat{\theta} = \theta_0 + \left( P_{\theta_0}^{-1} + \frac{1}{\sigma^2} \Phi' \Phi \right)^{-1} \Phi' \frac{1}{\sigma^2} (y - \Phi \theta_0) \quad (\text{D.182})$$

*Proof.* Introduce the function  $\tilde{L}(\theta)$

$$\begin{aligned} \tilde{L}(\theta) &= \frac{1}{2} \|\theta - \theta_0\|_{P_{\theta_0}^{-1}}^2 + \frac{1}{2} \frac{1}{\sigma^2} \|y - \Phi \theta\|_2^2 \\ &= \frac{1}{2} \theta' \left( P_{\theta_0}^{-1} + \frac{1}{\sigma^2} \Phi' \Phi \right) \theta - \left( P_{\theta_0}^{-1} \theta_0 + \frac{1}{\sigma^2} \Phi' y \right)' \theta \\ &\quad + \frac{1}{2} \|\theta_0\|_{P_{\theta_0}^{-1}}^2 + \frac{1}{2\sigma^2} \|y\|_2^2 \end{aligned} \quad (\text{D.183})$$

and note that  $L(\theta) = \alpha + \tilde{L}(\theta)$  in which  $\alpha$  is a constant independent of  $\theta$ . Consequently, the first order optimality conditions for (D.181) may be stated as

$$\nabla_{\theta} L(\hat{\theta}) = \nabla_{\theta} \tilde{L}(\hat{\theta}) = \left( P_{\theta_0}^{-1} + \frac{1}{\sigma^2} \Phi' \Phi \right) \hat{\theta} - \left( P_{\theta_0}^{-1} \theta_0 + \frac{1}{\sigma^2} \Phi' y \right) = 0 \quad (\text{D.184})$$

which implies

$$\begin{aligned} \hat{\theta} &= \left( P_{\theta_0}^{-1} + \frac{1}{\sigma^2} \Phi' \Phi \right)^{-1} \left( P_{\theta_0}^{-1} \theta_0 + \frac{1}{\sigma^2} \Phi' y \right) \\ &= \theta_0 + \left( P_{\theta_0}^{-1} + \frac{1}{\sigma^2} \Phi' \Phi \right)^{-1} \Phi' \frac{1}{\sigma^2} (y - \Phi \theta_0) \end{aligned} \quad (\text{D.185})$$

□

**Corollary D.2.16**

The estimate,  $\hat{\theta}$ , computed by (D.182) may be expressed as

$$\hat{\theta} = \theta_0 + P_{\theta_0} \Phi' (\sigma^2 I + \Phi P_{\theta_0} \Phi')^{-1} (y - \Phi \theta_0) \quad (\text{D.186})$$

*Proof.* Application of the matrix inversion lemma to  $(P_{\theta_0}^{-1} + \frac{1}{\sigma^2} \Phi' \Phi)^{-1} = (P_{\theta_0}^{-1} + \Phi' (\sigma^2 I)^{-1} \Phi)^{-1}$  gives the desired result. □

**Remark D.2.17**

The relation between the posterior estimate  $\hat{\theta}$  and the recursive maximum likelihood parameter estimation should be noted.

By the stochastic model (D.176) it may be noted that the joint distribution of  $\theta$  and  $y$  is

$$\begin{bmatrix} \theta \\ y \end{bmatrix} \sim N \left( \begin{bmatrix} \theta_0 \\ \Phi\theta_0 \end{bmatrix}, \begin{bmatrix} P_{\theta_0} & P_{\theta_0}\Phi' \\ \Phi P_{\theta_0} & \Phi P_{\theta_0}\Phi' + \sigma^2 I \end{bmatrix} \right) \quad (\text{D.187})$$

Consequently, the distribution of the conditional stochastic variable  $\theta|y$  is

$$\theta|y \sim N(\hat{\theta}, P) \quad (\text{D.188a})$$

in which the mean,  $\hat{\theta}$ , and covariance,  $P$ , are

$$\hat{\theta} = \theta_0 + P_{\theta_0}\Phi'(\sigma^2 I + \Phi P_{\theta_0}\Phi')^{-1}(y - \Phi\theta_0) \quad (\text{D.188b})$$

$$P = P_{\theta_0} - \Phi P_{\theta_0}(\sigma^2 I + \Phi P_{\theta_0}\Phi')^{-1}P_{\theta_0}\Phi' \quad (\text{D.188c})$$

In this particular case, the posterior mode and conditional mean are identical and it is possible to characterize the conditional distribution completely. In fact, this result is used extensively in one derivation of the Kalman filter.

Typically,  $n \gg p$  and the mean and covariance are most efficiently computed as

$$\hat{\theta} = \theta_0 + \left( P_{\theta_0}^{-1} + \frac{1}{\sigma^2} \Phi' \Phi \right)^{-1} \Phi' \frac{1}{\sigma^2} (y - \Phi\theta_0) \quad (\text{D.189a})$$

$$P = \left( P_{\theta_0}^{-1} + \frac{1}{\sigma^2} \Phi' \Phi \right)^{-1} \quad (\text{D.189b})$$

**D.2.5.2 Unknown Covariance**

Typically, when the covariance is unknown, the posterior estimate of the parameters,  $\theta$ , and the covariance,  $\sigma^2$ , in (D.176) is accomplished by computing the estimate,  $(\hat{\theta}, \hat{\sigma})$ , as the mode of the posterior distribution using a numerical procedure.

Hamilton (1994) provide an interesting example in which the posterior distribution can be characterized analytically. In this case the prior distribution of  $\theta$  and  $\sigma^2$  are

$$\sigma^{-2} \sim \Gamma(n_0, \lambda_0) \quad (\text{D.190a})$$

$$\theta|\sigma^{-2} \sim N(\theta_0, P_{\theta_0}) \quad (\text{D.190b})$$

The joint probability density function is  $p(\theta, \sigma^{-2}) = p(\theta|\sigma^{-2})p(\sigma^{-2})$ . For such a system an analytical characterization of the posterior distribution exists. We refer to proposition 12.3 in Hamilton (1994) for the resulting distribution.

## D.3 Multivariate Regression

Consider the stochastic linear model

$$\mathbf{y}_k = Ax_k + \mathbf{v}_k \quad \mathbf{v}_k \sim N_{iid}(0, R) \quad k = 1, 2, \dots, N \quad (\text{D.191})$$

in which  $\mathbf{y}_k : \Omega \mapsto \mathbb{R}^m$  and  $\mathbf{v}_k : \Omega \mapsto \mathbb{R}^m$  are stochastic variables,  $x_k \in \mathbb{R}^n$  are the regressors,  $A \in \mathbb{R}^{m \times n}$  denotes the parameters, and the covariance of the output errors is  $R$ . The experiment is conducted  $N$  times, such that the data available for estimation of the parameters are  $\{y_k, x_k\}_{k=1}^N$ .

The main assumptions invoked in this model are

1. The regressor sequence,  $\{x_k\}_{k=1}^N$ , is deterministic.
2.  $\{\mathbf{v}_k\}_{k=1}^N$  is a sequence of independent identically normal distributed variables with  $\mathbf{v}_k \sim N_{iid}(0, R)$ .

The model (D.191) may be expression in matrix notation

$$\mathbf{Y} = AX + \mathbf{V} \quad (\text{D.192})$$

in which

$$\mathbf{Y} = [\mathbf{y}_1 \quad \dots \quad \mathbf{y}_N] \quad (\text{D.193a})$$

$$X = [x_1 \quad \dots \quad x_N] \quad (\text{D.193b})$$

$$\mathbf{V} = [\mathbf{v}_1 \quad \dots \quad \mathbf{v}_N] \quad (\text{D.193c})$$

(D.192) may be expressed as

$$\text{vec}(\mathbf{Y}) = (X' \otimes I_m) \text{vec}(A) + \text{vec}(\mathbf{V}) \quad (\text{D.194})$$

which is equivalent with

$$\mathbf{y} = (X' \otimes I_m) a + \mathbf{v} \quad \mathbf{v} \sim N(0, I_N \otimes R) \quad (\text{D.195})$$

using the notation

$$\mathbf{y} = \text{vec}(\mathbf{Y}) \quad (\text{D.196a})$$

$$a = \text{vec}(A) \quad (\text{D.196b})$$

$$\mathbf{v} = \text{vec}(\mathbf{V}) \quad (\text{D.196c})$$

The predictors in (D.191), (D.192), and (D.195) are, respectively:

$$\hat{y}(x_k|A) = Ax_k \quad k = 1, 2, \dots, N \quad (\text{D.197a})$$

$$\hat{Y}(X|A) = AX \quad (\text{D.197b})$$

$$\hat{y}(X|a) = (X' \otimes I_m) a \quad (\text{D.197c})$$



### D.3.1 Least Squares Regression

The least squares criterion for estimating the parameters  $A$  is

$$\begin{aligned}\phi &= \frac{1}{2} \sum_{k=1}^N \|y_k - Ax_k\|_2^2 = \frac{1}{2} \sum_{k=1}^N (y_k - Ax_k)'(y_k - Ax_k) \\ &= \frac{1}{2} \|Y - AX\|^2\end{aligned}\quad (\text{D.198})$$

Note that in this expression, we have defined the norm,  $\|\cdot\| : \mathbb{R}^{m \times n} \mapsto \mathbb{R} \setminus \mathbb{R}_-$ , of a matrix,  $Z = \begin{bmatrix} z_1 & \dots & z_n \end{bmatrix} \in \mathbb{R}^{m \times n}$ , as

$$\|Z\|^2 = \sum_{k=1}^n \|z_k\|_2^2 = \sum_{k=1}^n z_k' z_k \quad (\text{D.199})$$

With this matrix norm definition, we have equivalence between this matrix norm and the two-norm for the case

$$\|Z\|^2 = \|\text{vec}(Z)\|_2^2 \quad \text{vec}(Z) = \begin{bmatrix} z_1 \\ \vdots \\ z_n \end{bmatrix} \quad (\text{D.200})$$

With this norm definition, the least squares problem may be defined as

$$\min_{A \in \mathbb{R}^{m \times n}} \phi = \frac{1}{2} \|Y - AX\|^2 \quad (\text{D.201})$$

and the least squares estimate is denoted

$$\hat{A} = \arg \min_{A \in \mathbb{R}^{m \times n}} \phi = \frac{1}{2} \|Y - AX\|^2 \quad (\text{D.202})$$

The least squares estimate for the model (D.191) stated in the matrix form (D.192) is stated in the following proposition.

**Proposition D.3.1 (Least squares estimate)**

Assume that  $(XX')$  is non-singular. The least squares estimate

$$\hat{A} = \arg \min_{A \in \mathbb{R}^{m \times n}} \phi = \frac{1}{2} \|Y - AX\|^2 \quad (\text{D.203})$$

of the model (D.192) is

$$\hat{A} = YX'(XX')^{-1} \quad (\text{D.204})$$

*Proof.* Notice

$$y = \text{vec}(Y) \quad (\text{D.205})$$

$$\text{vec}(AX) = (X' \otimes I_m) \text{vec}(A) = \Phi \theta \quad (\text{D.206})$$

in which

$$\Phi = X' \otimes I_m \quad (\text{D.207})$$

$$\theta = \text{vec}(A) \in \mathbb{R}^{mn} \quad (\text{D.208})$$

Consequently

$$\phi = \frac{1}{2} \|Y - AX\|^2 = \frac{1}{2} \|\text{vec}(Y - AX)\|_2^2 = \frac{1}{2} \|y - \Phi\theta\|_2^2 \quad (\text{D.209})$$

and

$$\begin{aligned} \text{vec}(\hat{A}) = \hat{\theta} &= \arg \min_{\theta} \left\{ \phi = \frac{1}{2} \|y - \Phi\theta\|_2^2 \right\} = (\Phi' \Phi)^{-1} \Phi' y \\ &= [(X' \otimes I_m)'(X' \otimes I_m)]^{-1} (X' \otimes I_m)' \text{vec}(Y) \\ &= [(XX') \otimes I_m]^{-1} (X \otimes I_m) \text{vec}(Y) \\ &= [(XX')^{-1} \otimes I_m] (X \otimes I_m) \text{vec}(Y) \\ &= [(XX')^{-1} X \otimes I_m] \text{vec}(Y) \\ &= \text{vec}(I_m Y X' (XX')^{-1}) \\ &= \text{vec}(Y X' (XX')^{-1}) \end{aligned} \quad (\text{D.210})$$

Hence

$$\hat{A} = Y X' (XX')^{-1} \quad (\text{D.211})$$

□

### Remark D.3.2

Define the sample covariances as

$$\hat{R}_{YX} = \frac{1}{N} \sum_{k=1}^N y_k x'_k = \frac{1}{N} Y X' \quad (\text{D.212})$$

$$\hat{R}_{XX} = \frac{1}{N} \sum_{k=1}^N x_k x'_k = \frac{1}{N} X X' \quad (\text{D.213})$$

Then the least squares estimate (D.204) may also be computed by

$$\hat{A} = Y X' (XX')^{-1} = \left( \frac{1}{N} Y X' \right) \left( \frac{1}{N} X X' \right)^{-1} = \hat{R}_{YX} \hat{R}_{XX}^{-1} \quad (\text{D.214})$$

■

### Remark D.3.3

From the least squares expression (D.204), it is evident that each row of  $A$  may be estimated separately, i.e.

$$\hat{A}_{i,:} = Y_{i,:} X' (XX')^{-1} \quad i = 1, 2, \dots, m \quad (\text{D.215})$$

This implies that the multivariate output problem can be solved by solving  $m$  univariate output problems. However, the advantage of the multivariate output formulation is that the matrix  $(XX')$  is inverted only once, while in a naive solution of  $m$  univariate output problems, it would be inverted  $m$  times.

For applications in system identification, this observation implies that there is no difference between the MIMO and MISO estimate, as long as the regressor matrix,  $X$ , is identical in the two cases. This implies that each individual output may be predicted independently. ■

**Proposition D.3.4 (Distribution of the least squares estimate)**

Let  $(XX')$  be non-singular. Then the least squares estimate (D.204) of  $A$  in the model (D.191) expressed in matrix form (D.192) has the distribution

$$\text{vec}(\hat{\mathbf{A}}) \sim N(\text{vec}(A), (XX')^{-1} \otimes R) \quad (\text{D.216})$$

*Proof.* The least squares estimate is a realization of

$$\begin{aligned} \text{vec}(\hat{\mathbf{A}}) &= [(XX')^{-1}X \otimes I_m] \text{vec}(\mathbf{Y}) \\ &= [(XX')^{-1}X \otimes I_m] [(X' \otimes I_m)\text{vec}(A) + \text{vec}(\mathbf{V})] \\ &= \text{vec}(A) + [(XX')^{-1}X \otimes I_m] \text{vec}(\mathbf{V}) \end{aligned} \quad (\text{D.217})$$

in which

$$\text{vec}(\mathbf{V}) = \mathbf{v} \sim N(0, I_N \otimes R) \quad (\text{D.218})$$

As  $\text{vec}(\hat{\mathbf{A}})$  is a linear combination of normally distributed variables,  $\text{vec}(\hat{\mathbf{A}})$  is also normally distributed.  $\text{vec}(\hat{\mathbf{A}})$  has mean

$$E\{\text{vec}(\hat{\mathbf{A}})\} = \text{vec}(A) \quad (\text{D.219})$$

and covariance

$$\begin{aligned} V\{\text{vec}(\hat{\mathbf{A}})\} &= \langle \text{vec}(\hat{\mathbf{A}}) - \text{vec}(A), \text{vec}(\hat{\mathbf{A}}) - \text{vec}(A) \rangle \\ &= \langle [(XX')^{-1}X \otimes I_m] \mathbf{v}, [(XX')^{-1}X \otimes I_m] \mathbf{v} \rangle \\ &= [(XX')^{-1}X \otimes I_m] \langle \mathbf{v}, \mathbf{v} \rangle [(XX')^{-1}X \otimes I_m]' \\ &= [(XX')^{-1}X \otimes I_m] (I_N \otimes R) [X'(XX')^{-1} \otimes I_m] \\ &= [(XX')^{-1}X \otimes R] [X'(XX')^{-1} \otimes I_m] \\ &= [(XX')^{-1}XX'(XX')^{-1}] \otimes R \\ &= (XX')^{-1} \otimes R \end{aligned} \quad (\text{D.220})$$

□

**Remark D.3.5**

The distribution (D.216) of the least squares estimate may also be expressed as

$$\frac{1}{\sqrt{N}} \text{vec}(\hat{\mathbf{A}} - A) \sim N(0, \hat{R}_{XX}^{-1} \otimes R) \quad (\text{D.221})$$

■

With the estimate,  $\hat{A}$ , given, the noise in the models (D.191) and (D.192) may be estimated as

$$\hat{v}_k = y_k - \hat{A}x_k \quad k = 1, 2, \dots, N \quad (\text{D.222a})$$

$$\hat{V} = Y - \hat{A}X \quad (\text{D.222b})$$

These expressions can be used for estimating the covariance of the noise. The maximum likelihood covariance estimate is

$$\hat{R} = \frac{1}{N} \hat{V} \hat{V}' = \frac{1}{N} \sum_{k=1}^N \hat{v}_k \hat{v}_k' \quad (\text{D.223})$$

while an unbiased estimate of the covariance may be computed as

$$\hat{R} = \frac{1}{N-n} \hat{V} \hat{V}' = \frac{1}{N-n} \sum_{k=1}^N \hat{v}_k \hat{v}_k' \quad (\text{D.224})$$

In these expressions, we have assumed that the regression is constructed such that the estimated noise has zero mean.

The true noise covariance,  $R$ , is usually not available. Hence, it is natural to replace it with the estimated noise covariance,  $\hat{R}$ , in the covariance of the parameter estimates, i.e.

$$\text{vec}(\hat{\mathbf{A}}) \sim N(\text{vec}(A), (XX')^{-1} \otimes \hat{R}) \quad (\text{D.225})$$

### D.3.2 Maximum Likelihood Regression

The likelihood function of an observation,  $y_k$ , given  $A$  and  $R$  in the model (D.191) is

$$\begin{aligned} p(y_k|A, R) &= p(v_k) = (2\pi)^{-m/2} (\det R)^{-1/2} \exp \left( -\frac{1}{2} v_k' R^{-1} v_k \right) \\ &= (2\pi)^{-m/2} (\det R)^{-1/2} \exp \left( -\frac{1}{2} (y_k - Ax_k)' R^{-1} (y_k - Ax_k) \right) \end{aligned} \quad (\text{D.226})$$

for  $k = 1, 2, \dots, N$ . The independence of each observation  $y_k$  from the other observations implies that the likelihood function,  $p(Y|A, R)$ , for the model (D.191) is

$$\begin{aligned} p(Y|A, R) &= \prod_{k=1}^N p(y_k|A, R) \\ &= (2\pi)^{-mN/2} (\det R)^{-N/2} \exp \left( -\frac{1}{2} \sum_{k=1}^N (y_k - Ax_k)' R^{-1} (y_k - Ax_k) \right) \end{aligned} \quad (\text{D.227})$$

$$L(A, R; Y) = -\ln p(Y|A, R)$$

$$\begin{aligned}
&= \frac{Nm}{2} \ln(2\pi) + \frac{N}{2} \ln(\det R) + \frac{1}{2} \sum_{k=1}^N (y_k - Ax_k)' R^{-1} (y_k - Ax_k) \\
&= \frac{Nm}{2} \ln(2\pi) + \frac{N}{2} \ln(\det R) + \frac{1}{2} \sum_{k=1}^N \text{tr} [R^{-1} (y_k - Ax_k)(y_k - Ax_k)'] \\
&= \frac{Nm}{2} \ln(2\pi) + \frac{N}{2} \ln(\det R) + \frac{1}{2} \text{tr} \left[ R^{-1} \sum_{k=1}^N (y_k - Ax_k)(y_k - Ax_k)' \right] \\
&= \frac{Nm}{2} \ln(2\pi) + \frac{N}{2} \ln(\det R) + \frac{1}{2} \text{tr} [R^{-1} (Y - AX)(Y - AX)']
\end{aligned} \tag{D.228}$$

$$\begin{aligned}\hat{A} &= \arg \max_{A \in \mathbb{R}^{m \times n}} p(Y|A, R) \\ &= \arg \min_{A \in \mathbb{R}^{m \times n}} L(A, R; Y)\end{aligned}\tag{D.229}$$
$$\begin{aligned} (\hat{A}, \hat{R}) &= \arg \max_{A \in \mathbb{R}^{m \times n}} p(Y|A, R) \\ &= \arg \min_{A \in \mathbb{R}^{m \times n}} L(A, R; Y) \end{aligned} \quad (\text{D.230})$$

Let  $A$ ,  $B$ , and  $D$  be compatible real matrices. Then

$$\frac{\partial}{\partial A} \text{tr}(BAD) = B'D' \quad (\text{D.232})$$

$$\frac{\partial}{\partial A} \text{tr}(BADA') = B'AD' + BAD \quad (\text{D.233})$$

$$\frac{\partial}{\partial A} \text{tr}(BADA') = 2BAD \quad (\text{D.234})$$

*Proof.* Bard (1974) provides the technique for developing matrix derivatives. Poulsen (1995c) provides some of the results stated.  $\square$

**Lemma D.3.7**

Let  $A \in \mathbb{R}^{n \times n}$  and  $B \in \mathbb{R}^{n \times n}$  be symmetric positive definite matrices. Then

$$\frac{\partial}{\partial A} \ln \det A = A^{-1} \quad (\text{D.235})$$

$$\frac{\partial}{\partial A} \text{tr}(A^{-1}B) = -A^{-1}BA^{-1} \quad (\text{D.236})$$

*Proof.* See Bard (1974).  $\square$

**Lemma D.3.8**

The matrix derivatives of the negative log likelihood function (D.228) with respect to  $A \in \mathbb{R}^{m \times n}$  and  $R \in \mathbb{R}^{m \times m}$  are, respectively:

$$\frac{\partial L}{\partial A} = -R^{-1}YX' + R^{-1}AXX' \quad (\text{D.237})$$

$$\frac{\partial L}{\partial R} = \frac{N}{2}R^{-1} - \frac{1}{2}R^{-1}(Y - AX)(Y - AX)'R^{-1} \quad (\text{D.238})$$

*Proof.* The negative log likelihood function is

$$L(A, R; Y) = \frac{Nm}{2} \ln(2\pi) + \frac{N}{2} \ln(\det R) + \frac{1}{2} \text{tr} \left( R^{-1}(Y - AX)(Y - AX)' \right) \quad (\text{D.239})$$

Notice

$$\begin{aligned} R^{-1}(Y - AX)(Y - AX)' &= \\ R^{-1}YY' - R^{-1}YX'A' - R^{-1}AXY' + R^{-1}AXX'A' \end{aligned} \quad (\text{D.240})$$

Consequently, application of lemma D.3.6 gives

$$\begin{aligned} \frac{\partial L}{\partial A} &= \frac{1}{2} \frac{\partial}{\partial A} \text{tr} \left( R^{-1}(Y - AX)(Y - AX)' \right) \\ &= \frac{1}{2} \frac{\partial}{\partial A} \text{tr} \left( R^{-1}YY' \right) - \frac{1}{2} \frac{\partial}{\partial A} \text{tr} \left( R^{-1}YX'A' \right) \\ &\quad - \frac{1}{2} \frac{\partial}{\partial A} \text{tr} \left( R^{-1}AXY' \right) + \frac{1}{2} \frac{\partial}{\partial A} \text{tr} \left( R^{-1}AXX'A' \right) \\ &= 0 - \frac{1}{2} R^{-1}YX' - \frac{1}{2} R^{-1}(XY')' + \frac{1}{2} 2R^{-1}AXX' \\ &= -R^{-1}YX' + R^{-1}AXX' \end{aligned} \quad (\text{D.241})$$

Similarly, application of lemma D.3.7 yields

$$\begin{aligned} \frac{\partial L}{\partial R} &= \frac{N}{2} \frac{\partial}{\partial R} \ln(\det R) + \frac{1}{2} \frac{\partial}{\partial R} \text{tr} \left( R^{-1}(Y - AX)(Y - AX)' \right) \\ &= \frac{N}{2} R^{-1} - \frac{1}{2} R^{-1}(Y - AX)(Y - AX)'R^{-1} \end{aligned} \quad (\text{D.242})$$

$\square$

**Proposition D.3.9 (ML estimate with known covariance)**

Let  $XX'$  be non-singular. The maximum likelihood estimate

$$\hat{A} = \arg \min_{A \in \mathbb{R}^{m \times n}} L(A, R; Y) \quad (\text{D.243})$$

of  $A$  in the model (D.191) with known noise covariance,  $R$ , is

$$\hat{A} = YX'(XX')^{-1} \quad (\text{D.244})$$

*Proof.* The maximum likelihood problem is convex (Lütkepohl, 1993). The first order necessary and sufficient optimality condition for the considered maximum likelihood problem is

$$\frac{\partial L}{\partial A} = -R^{-1}YX' + R^{-1}\hat{A}XX' = 0 \quad (\text{D.245})$$

which implies

$$R^{-1}(\hat{A}XX' - YX') = 0 \quad \Leftrightarrow \quad \hat{A} = YX'(XX')^{-1} \quad (\text{D.246})$$

□

**Proposition D.3.10 (ML estimate with unknown covariance)**

Let  $XX'$  be non-singular. The maximum likelihood estimate

$$(\hat{A}, \hat{R}) = \arg \min_{A \in \mathbb{R}^{m \times n}, R \in \mathbb{R}^{m \times m}} L(A, R; Y) \quad (\text{D.247})$$

of  $A$  and  $R$  in the model (D.191) with unknown noise covariance,  $R$ , is

$$\hat{A} = YX'(XX')^{-1} \quad (\text{D.248a})$$

$$\hat{R} = \frac{1}{N}(Y - \hat{A}X)(Y - \hat{A}X)' \quad (\text{D.248b})$$

*Proof.* The maximum likelihood problem is convex (Lütkepohl, 1993). The first order necessary and sufficient optimality conditions for the considered maximum likelihood problem are

$$\frac{\partial L}{\partial A} = -\hat{R}^{-1}YX' + \hat{R}^{-1}\hat{A}XX' = 0 \quad (\text{D.249a})$$

$$\frac{\partial L}{\partial R} = \frac{N}{2}\hat{R}^{-1} - \frac{1}{2}\hat{R}^{-1}(Y - \hat{A}X)(Y - \hat{A}X)'\hat{R}^{-1} = 0 \quad (\text{D.249b})$$

(D.249a) implies

$$\hat{R}^{-1}(\hat{A}XX' - YX') = 0 \quad \Leftrightarrow \quad \hat{A} = YX'(XX')^{-1} \quad (\text{D.250})$$

and (D.249b) yields

$$\frac{1}{2}\hat{R}^{-1}(NI - (Y - \hat{A}X)(Y - \hat{A}X)'\hat{R}^{-1}) = 0 \quad \Leftrightarrow \quad (\text{D.251a})$$

$$\hat{R} = \frac{1}{N}(Y - \hat{A}X)(Y - \hat{A}X)' \quad (\text{D.251b})$$

□

# E

## Prediction-Error-Methods for Model Predictive Control

Prediction-error-methods tailored for state space model based predictive control are presented. The prediction-error methods studied are based on predictions using the Kalman filter and predictors for a linear discrete-time stochastic state space model. Both single-step and multi-step prediction-error methods based on least squares, maximum likelihood and maximum a posteriori criteria are derived and presented. It is argued that the prediction-error criterion should be selected such that it is compatible with the objective function of the predictive controller in which the model is to be applied. Realization of the discrete-time stochastic state space model from a continuous-discrete-time linear stochastic system specified using transfer functions with time-delays is outlined. The proposed prediction error-methods are demonstrated for a SISO system parameterized by the transfer functions with time delays of a continuous-discrete-time linear stochastic system. The simulations for this case suggest to use the one-step-ahead prediction-error maximum-likelihood (or maximum a posteriori) estimator. It gives consistent estimates of all parameters and the parameter estimates are almost identical to the estimates obtained for long prediction horizons but with consumption of significantly less computational resources. The suitability of the proposed method for predictive control is demonstrated for dual composition control of a simulated binary distillation column.

### E.1 Introduction

Predictive control computes the controls based on a prediction of the outputs. The predictions are based on a model of the system. Original model predictive control algorithms developed for the process industries such as *Dynamic Matrix Control* (Cutler and Ramaker, 1980) and *Model Predictive Heuristic Control* (Richalet *et al.*, 1978) are based on step and impulse response models for the



output predictions and introduce feedback by updating a bias term representing an output disturbance. In contrast to the approach applied by the process industries, the academic system identification community developed predictive controllers based on ARMAX (ARIMAX, CARIMA) models (Box *et al.*, 1994; Hallager *et al.*, 1984; Clarke *et al.*, 1987a; Bitmead *et al.*, 1990). *Generalized Predictive Control* (Clarke *et al.*, 1987a) is the most famous implementation of this class of predictive controllers that use an ARMAX model for the filter in generating the feedback as well as for the output predictions. The success of these model based control algorithms and their applications is to a large extent due to existence of efficient methods for generation of the models needed by these controllers.

Modern predictive control is discussed in terms of state space models (Muske and Rawlings, 1993a; Rawlings, 2000; Maciejowski, 2002). These algorithms are based on a discrete-time linear stochastic state space model. They apply a Kalman filter to compute the current state based on the measurements, and a Kalman predictor for predicting the outputs (Kailath *et al.*, 2000). State space based model predictive controllers may be derived by realization of input-output models (i.e. FIR, ARX, ARMAX models) in state space form. However, no advice has been given regarding direct construction of a linear state space model suitable for predictive control. Therefore, better identification methods for state space based predictive control is requested (Morari and Lee, 1999; Jørgensen and Lee, 2001; Hjalmarsson, 2003; Gevers, 2003, 2004).

This paper addresses construction of stochastic linear state space models using the *prediction-error-method* (Åström, 1980; Ljung, 1999; Kristensen *et al.*, 2004c). In particular, we argue that for predictive control the parameters in the model should be based on the multi-step prediction error compatible with the predictive controller in which the predictor is to be used. This approach differs from the standard prediction-error-method in which the single-step one-step-ahead prediction error is minimized. Shah and coworkers (Shook *et al.*, 1992; Gopaluni *et al.*, 2003, 2004) apply a similar multi-step approach based on impulse response models and a least-squares criterion. The approach presented in this paper distinguishes itself by being general for linear systems, by applying least-squares as well as maximum likelihood criteria for the prediction errors in the estimator, and in particular by being directly applicable to state space model based predictive control in its modern implementation.

In section E.2, the optimal filter and predictor for a discrete time linear system in state space form is presented and the application of this predictor in a predictive controller is stated. Section E.3 introduces the prediction error of the predictor and demonstrates how the prediction error may be applied in estimating the parameters in the model - or rather the parameters in the predictor - using the prediction error method. The least squares criterion, the maximum likelihood criterion and the maximum a posteriori criterion are discussed for the single-step one-step-ahead and  $j$ -step-ahead prediction error as well as for the multi-step prediction error. It is argued that for predictive control, the

multi-step prediction error should be used. Section E.4 introduces continuous-discrete-time systems in which the continuous part consists of a system of stochastic differential equations. Transformation of this system to a discrete-time system is presented. A continuous-discrete-time deterministic-stochastic transfer function model with time delays is stated and it is outlined how the discretization of linear stochastic differential equations may be used for discrete-time realization of the continuous-discrete-time deterministic-stochastic transfer function model with time delays. By this discrete-time realization technique the prediction error methods can be applied to continuous-discrete-time deterministic-stochastic transfer function models with time delays. Section E.5 and E.6 demonstrate the methodology for a SISO and MIMO system, respectively. The MIMO system represents a high purity binary distillation column and the performance of a constrained predictive controller based on the estimated model is demonstrated. Finally, in section E.7 conclusions are drawn.

## E.2 Predictive Control

The predictive controller considered assumes that the system,  $\mathcal{S}$ , can be described by a linear stochastic discrete-time difference equation of the form

$$\mathbf{x}_{k+1} = A(\theta)\mathbf{x}_k + B(\theta)u_k + \mathbf{w}_k \quad (\text{E.1a})$$

$$\mathbf{y}_k = C(\theta)\mathbf{x}_k + \mathbf{v}_k \quad (\text{E.1b})$$

in which

$$\begin{bmatrix} \mathbf{w}_k \\ \mathbf{v}_k \end{bmatrix} \sim N_{iid} \left( \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} R_{ww}(\theta) & R_{wv}(\theta) \\ R_{wv}(\theta)' & R_{vv}(\theta) \end{bmatrix} \right) \quad (\text{E.1c})$$

and

$$\mathbf{x}_0 \sim N(\hat{x}_{0|-1}(\theta), P_{0|-1}(\theta)) \quad (\text{E.1d})$$

The system matrices,  $(A = A(\theta), B = B(\theta), C = C(\theta))$ , are parameterized in terms of the parameter vector,  $\theta$ . This parameter vector is also used to specify the distribution of the exogenous stochastic variables, i.e. to specify the covariance matrices,  $(R_{ww} = R_{ww}(\theta), R_{wv} = R_{wv}(\theta), R_{vv} = R_{vv}(\theta))$ , of the noise terms as well as to specify the initial condition,  $\hat{x}_{0|-1} = \hat{x}_{0|-1}(\theta)$ , and its covariance  $P_{0|-1} = P_{0|-1}(\theta)$ . The states,  $\mathbf{x}_k$ , the process noise,  $\mathbf{w}_k$ , the measurement noise,  $\mathbf{v}_k$ , and the outputs,  $\mathbf{y}_k$ , are stochastic vectors. As  $\mathbf{x}_0$ ,  $\mathbf{w}_k$ , and  $\mathbf{v}_k$  are normally distributed and the system is linear, the states,  $\mathbf{x}_k$ , and the outputs,  $\mathbf{y}_k$  are also normally distributed. The inputs,  $u_k$ , are assumed to be deterministic and in particular assumed to be uncorrelated with the process measurements,  $\mathbf{y}_k$ . This assumption implies that the IO-data,  $\{y_k, u_k\}_{k=0}^{N-1}$ , are collected for a system that operates in *open-loop*.

Assume that the system matrices, noise covariance matrices, and distribution of the model,  $\mathcal{M}$ , and the true system,  $\mathcal{S}$ , are identical. Then prediction uncertainties and errors are due to the stochastic nature of the initial state,

the process noise, and the measurement noise, but not due to any systematic trend as a consequence of an incorrect model. In this case, the optimal filter and predictor is the Kalman filter and predictor (Kailath *et al.*, 2000). Under the same assumptions the *separation principle* is valid, and the optimal controller for the system can be split into an estimator and a *certainty equivalence* regulator (Bertsekas, 1995a). Obviously, the true system and the model are never identical. The separation principle does not hold either, as the system is constrained. Nevertheless, predictive control use the Kalman filter feedback, the Kalman predictor for the output predictions, and separates the controller into an estimator and a regulator. To guarantee offset free control in the case of model-plant mismatch as well as unknown disturbances, the model must be augmented with integrators (Muske and Badgwell, 2002; Pannocchia and Rawlings, 2003). For ARMAX models this is achieved by differencing the inputs and outputs (Hallager *et al.*, 1984; Clarke *et al.*, 1987a). In the face of these approximations and deliberate model modifications introduced to obtain steady-state offset-free control, it is clear that the structure of the estimated model,  $\mathcal{M}$ , in general will be different from the structure of true system,  $\mathcal{S}$ . Therefore, it seems reasonable to side with Ljung (Ljung, 1999) and view the parameter-estimation purpose as to obtain good predictors for the predictive controller rather than accurate parameters in the true unknown model.

### E.2.1 Filter and Predictor

The filter and predictor used in the predictive controller for the system (E.1) is the Kalman filter and predictor. The recursions defining the Kalman filter and predictor along with their covariances are stated in this subsection. The filter and predictors are the conditional states,  $\mathbf{x}_{k+j}|\mathcal{I}_k$ , and the conditional outputs,  $\mathbf{y}_{k+j}|\mathcal{I}_k$ , given the information vector  $\mathcal{I}_k$  defined recursively as  $\mathcal{I}_k = \{\mathcal{I}_{k-1}, y_k, u_{k-1}\}$ ,  $\mathcal{I}_0 = \{y_0\}$ , and  $\mathcal{I}_{-1} = \emptyset$ . As all states,  $\mathbf{x}_k$ , and outputs,  $\mathbf{y}_k$ , are normally distributed, the conditional states,  $\mathbf{x}_{k+j}|\mathcal{I}_k$ , and the conditional outputs,  $\mathbf{y}_{k+j}|\mathcal{I}_k$ , are also normally distributed. Normally distributed stochastic variables are completely characterized by their mean and covariance.

The Kalman filter and predictor algorithm stated next is the measurement-time updated Kalman filter and predictor (Kailath *et al.*, 2000). As new information,  $y_k$ , becomes available, the gains,  $K_{fx,k}$  and  $K_{fw,k}$ , and one-step prediction error,  $e_k$ , of the filter are updated according to

$$\hat{\mathbf{y}}_{k|k-1} = C\hat{\mathbf{x}}_{k|k-1} \quad (\text{E.2a})$$

$$e_k = y_k - \hat{\mathbf{y}}_{k|k-1} \quad (\text{E.2b})$$

$$R_{e,k} = CP_{k|k-1}C' + R_{vv} \quad (\text{E.2c})$$

$$K_{fx,k} = P_{k|k-1}C'R_{e,k}^{-1} \quad (\text{E.2d})$$

$$K_{fw,k} = R_{wv}R_{e,k}^{-1} \quad (\text{E.2e})$$

The filtered state and filtered process disturbance are normally distributed, i.e.

$\mathbf{x}_k|\mathcal{I}_k \sim N(\hat{x}_{k|k}, P_{k|k})$  and  $\mathbf{w}_k|\mathcal{I}_k \sim N(\hat{w}_{k|k}, Q_{k|k})$ . The expressions for the filtered conditional means are

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_{fx,k}e_k \quad (\text{E.3a})$$

$$\hat{w}_{k|k} = K_{fw,k}e_k \quad (\text{E.3b})$$

and the expressions for the filtered conditional covariances are

$$P_{k|k} = P_{k|k-1} - K_{fx,k}R_{e,k}K'_{fx,k} \quad (\text{E.4a})$$

$$Q_{k|k} = R_{ww} - K_{fw,k}R_{e,k}K'_{fw,k} \quad (\text{E.4b})$$

The one-step-ahead prediction of the state,  $\mathbf{x}_{k+1}|\mathcal{I}_k \sim N(\hat{x}_{k+1|k}, P_{k+1|k})$ , and the measured output,  $\mathbf{y}_{k+1}|\mathcal{I}_k \sim N(\hat{y}_{k+1|k}, R_{k+1|k})$ , are also normally distributed. The one-step-ahead prediction conditional means are

$$\hat{x}_{k+1|k} = A\hat{x}_{k|k} + B\hat{u}_{k|k} + \hat{w}_{k|k} \quad (\text{E.5a})$$

$$\hat{y}_{k+1|k} = C\hat{x}_{k+1|k} \quad (\text{E.5b})$$

and the associated covariances are computed by

$$P_{k+1|k} = AP_{k|k}A' + Q_{k|k} - AK_{fx,k}R'_{wv} - R_{wv}K'_{fx,k}A' \quad (\text{E.6a})$$

$$R_{k+1|k} = CP_{k+1|k}C' + R_{vv} \quad (\text{E.6b})$$

Similarly, the  $j$ -step-ahead predictions ( $j > 1$ ) of the states,  $\mathbf{x}_{k+j}|\mathcal{I}_k \sim N(\hat{x}_{k+j|k}, P_{k+j|k})$ , and the outputs,  $\mathbf{y}_{k+j}|\mathcal{I}_k \sim N(\hat{y}_{k+j|k}, R_{k+j|k})$ , are normally distributed. The  $j$ -step-ahead prediction conditional means are

$$\hat{x}_{k+j|k} = A\hat{x}_{k+j-1|k} + B\hat{u}_{k+j-1|k} \quad (\text{E.7a})$$

$$\hat{y}_{k+j|k} = C\hat{x}_{k+j|k} \quad (\text{E.7b})$$

and the associated conditional covariances are computed by

$$P_{k+j|k} = AP_{k+j-1|k}A' + R_{ww} \quad (\text{E.8a})$$

$$R_{k+j|k} = CP_{k+j|k}C' + R_{vv} \quad (\text{E.8b})$$

The recursions (E.2)-(E.8) specifies the Kalman filter and predictor equations used by the predictive controller completely. Often the  $j$ -step-ahead ( $j > 1$ ) conditional covariance equations (E.8) are not used directly in the predictive controller. Equations (E.2) and (E.3) are used in the estimator part for forming the filtered states and filtered process disturbances. Equations (E.4), (E.5a), and (E.6a) are used in the estimator part for updating the Kalman filter.

## E.2.2 Regulator

Given the conditional mean of the filtered state,  $\hat{x}_{k|k}$ , and the conditional mean of the filtered process disturbance,  $\hat{w}_{k|k}$ , the *certainty equivalence* predictive

regulator applies equations (E.5) and (E.7) for predicting the mean of the conditional outputs,  $\hat{y}_{k+j|k}$ , in the regulator objective function

$$\begin{aligned} \phi_k = & \frac{1}{2} \sum_{j=1}^{N_p} (\hat{y}_{k+j|k} - r_{k+j|k})' Q (\hat{y}_{k+j|k} - r_{k+j|k}) \\ & + \frac{1}{2} \sum_{j=0}^{N_c} \Delta \hat{u}'_{k+j|k} S \Delta \hat{u}_{k+j|k} \end{aligned} \quad (\text{E.9})$$

The objective,  $\phi_k$ , has a finite prediction horizon,  $N_p$ , and computes the optimal predicted inputs,  $\{\hat{u}_{k+j|k}\}_{j=0}^{N_p-1} = \{\hat{u}_{k+j|k}\}_{j=0}^{N_c} \cup \{\hat{u}_{k+j|k}\}_{j=N_c+1}^{N_p-1}$  by solving

$$\min_{\{\hat{u}_{k+j|k}\}_{j=0}^{N_p-1}} \phi_k \quad (\text{E.10a})$$

$$s.t. \quad (E.5), (E.7) \quad (\text{E.10b})$$

$$\hat{u}_{k+j|k} \in \mathbb{U}(\hat{x}_{k|k}, \hat{w}_{k|k}) \quad (\text{E.10c})$$

$$\hat{y}_{k+j|k} \in \mathbb{Y}(\hat{x}_{k|k}, \hat{w}_{k|k}) \quad (\text{E.10d})$$

and using some parametrization of the tail inputs,  $\{\hat{u}_{k+j|k}\}_{j=N_c+1}^{N_p-1}$  (Garcia *et al.*, 1989; Muske and Rawlings, 1993a; Scokaert and Rawlings, 1998; Mayne *et al.*, 2000; Maciejowski, 2002). The sets  $\mathbb{U}(\cdot, \cdot)$  and  $\mathbb{Y}(\cdot, \cdot)$  denote input constraints, rate of movement input constraints, and output constraints. (E.10) is a quadratic program with special structure for which efficient solution algorithms exist (Jørgensen *et al.*, 2004). Predictive control is implemented in a moving horizon manner, which means that the first optimal control,  $\hat{u}_{k|k}$ , of the optimal control sequence obtained by solving (E.10),  $\{\hat{u}_{k+j|k}\}_{j=0}^{N_p-1}$ , is implemented on the process, i.e.  $u_k = \hat{u}_{k|k}$ . In some implementations, the setup is modified slightly to accommodate the duration of computation and the implemented process input is  $u_k = \hat{u}_{k|k-1}$ .

The implemented process input,  $u_k = \hat{u}_{k|k}$ , is an implicit function of the output predictions. The objective function in the regulator of the model predictive controller requires multi-step output prediction, i.e.  $\hat{y}_{k+j|k}$  for  $j = 1, 2, \dots, N_p$ . Hence, intuitively it seems natural to select the model parameters based on multi-step prediction capabilities compatible with the regulator objective (Söderström and Stoica, 1989; Stoica and Nehorai, 1989; Shook *et al.*, 1992; Gopaluni *et al.*, 2003, 2004). This is in contrast to the usual approach in which the parameters are determined based on their single-step one-step-ahead prediction capabilities. In yet another alternative,  $N_p$  different models are identified, i.e. one model for each single-step  $j$ -step-ahead output prediction,  $\hat{y}_{k+j|k}$  for  $j = 1, 2, \dots, N_p$ . This implies that instead of using  $\hat{y}_{k+j|k}(\theta)$  for  $j = 1, 2, \dots, N_p$ , the predictors  $\hat{y}_{k+j|k}(\theta_j)$  for  $j = 1, 2, \dots, N_p$  are proposed. This multi-model approach has been applied for ARX models (Rossiter and Kouvaritakis, 2001; Haber *et al.*, 2003) and is also adopted in the MUSMAR

(multi-step multi-variable adaptive regulator) algorithm for predictive control (Greco *et al.*, 1984; Mosca, 1995).

## E.3 Prediction-Error-Methods

### E.3.1 Standard Regression Problem

The essence of regression is to select some parameters,  $\theta$ , such that the predicted outputs,  $\hat{y}_k(\theta)$ , match the measured outputs,  $y_k$ , as well as possible for all measurements  $k = 0, 1, \dots, N-1$ . The estimation problem is often stated as the stochastic relation

$$\mathbf{y}_k = \hat{\mathbf{y}}_k(\theta) + \mathbf{e}_k, \mathbf{e}_k \sim N(0, R_k), k = 0, 1, \dots, N-1 \quad (\text{E.11})$$

The predictor or estimator,  $\hat{\mathbf{y}}_k(\theta)$ , is a function of the parameters,  $\theta \in \Theta \subset \mathbb{R}^{n_\theta}$ . For the measured realization,  $\{y_k\}_{k=0}^{N-1}$ , of the outputs,  $\{\mathbf{y}_k\}_{k=0}^{N-1}$ , the parameters,  $\theta$ , are computed such that some measure, e.g. the least squares measure, of the residuals,  $\{e_k(\theta) = y_k - \hat{y}_k(\theta)\}_{k=0}^{N-1}$ , is minimized. This is the standard nonlinear regression problem (Seber and Wild, 1989; Hamilton, 1994), which can be stated as the optimization problem

$$\hat{\theta} = \arg \min_{\theta \in \Theta} V(\theta) \quad (\text{E.12})$$

with the objective function  $V(\theta) = V_{LS}(\theta)$  being

$$V_{LS}(\theta) = \frac{1}{2} \sum_{k=0}^{N-1} \|e_k(\theta)\|_2^2 \quad (\text{E.13})$$

in the least squares case. The maximum-likelihood estimate corresponds to using negative log-likelihood function in (E.12), i.e.  $V(\theta) = V_{ML}(\theta)$  with  $V_{ML}(\theta)$  defined as

$$\begin{aligned} V_{ML}(\theta) = & \frac{N n_y}{2} \ln(2\pi) + \frac{1}{2} \sum_{k=0}^{N-1} \ln(\det R_k(\theta)) \\ & + \frac{1}{2} \sum_{k=0}^{N-1} e_k(\theta)' R_k(\theta)^{-1} e_k(\theta) \end{aligned} \quad (\text{E.14})$$

The maximum *a posteriori* estimate assumes that a priori the parameters stem from the distribution  $\boldsymbol{\theta} \sim N(\theta_0, P_{\theta_0})$  in which  $\theta_0 \in \Theta \subset \mathbb{R}^{n_\theta}$ . Then using Bayes rule the negative log-likelihood *a posteriori* function is

$$\begin{aligned} V_{MAP}(\theta) = & V_{ML}(\theta) + \frac{n_\theta}{2} \ln(2\pi) + \frac{1}{2} \ln(\det P_{\theta_0}) \\ & + \frac{1}{2} (\theta - \theta_0)' P_{\theta_0}^{-1} (\theta - \theta_0) \end{aligned} \quad (\text{E.15})$$

Hence, the maximum *a posteriori* estimate is obtained by applying  $V(\theta) = V_{MAP}(\theta)$  in (E.12).

### E.3.2 The PE Method as a Regression Problem

The family of prediction-error-methods can be considered as solving a general regression problem similar to (E.11). The estimate of the prediction error estimates are obtained by solving an optimization problem like (E.12) for some criteria (LS, ML, MAP) and some predictors. For the case considered in this paper, the predictors in the prediction error method are the Kalman predictors,  $\hat{y}_{k+j|k}(\theta)$ . The prediction errors,  $\varepsilon_{k+j|k} = y_{k+j} - \hat{y}_{k+j|k}(\theta)$ , correspond to the residuals in the standard regression problem. Therefore, the prediction-error-method is a standard regression problem with a predictor generated by the Kalman filter and predictor. In the following, the statistical properties of the predictors and the prediction errors will be discussed and various criteria for estimating the parameters in the prediction error framework are presented.

If it is possible to know the true structure of the system,  $\mathcal{S}$ , and the model identified,  $\mathcal{M}(\theta)$ , is equal to the true system,  $\mathcal{M}(\theta) = \mathcal{S}$ , then this model will be optimal in a statistical sense no matter for what purpose it is to be used and what consistent estimator (criterion) used for determining the parameters. In any realistic situation, it is almost impossible to know the true model structure due to changing process conditions, changing disturbance properties and nonlinearities. Therefore, in practice the model should be suited and be identified for the purpose it is going to be used. In predictive control this corresponds to minimization of multi-step predictions compatible with the regulator objective function.

### E.3.3 Single-Step j-Step-Ahead Prediction Error

Let the time indices be  $k = -1, 0, 1, \dots, N-1-j$  and the prediction index be  $1 \leq j \leq N_p$ . This implies that  $0 \leq k+j \leq N-1$ . The conditional outputs,  $\mathbf{y}_{k+j}|\mathcal{I}_k$ , have the distribution

$$\mathbf{y}_{k+j}|\mathcal{I}_k \sim N(\hat{\mathbf{y}}_{k+j|k}, R_{k+j|k}) \quad (\text{E.16})$$

and their correlation may be computed by (Kailath *et al.*, 2000)

$$\begin{aligned} R_{(i,j)|k} &= \langle (\mathbf{y}_{k+i}|\mathcal{I}_k) - \hat{\mathbf{y}}_{k+i|k}, (\mathbf{y}_{k+j}|\mathcal{I}_k) - \hat{\mathbf{y}}_{k+j|k} \rangle \\ &= \begin{cases} CA^{i-j-1}N_{k+j|k} & i > j \\ CP_{k+i|k}C' + R_{vv} & i = j \\ N'_{k+i|k}(A^{j-i-1})'C' & i < j \end{cases} \end{aligned} \quad (\text{E.17})$$

in which  $1 \leq i \leq N_p$ ,  $1 \leq j \leq N_p$ , and

$$N_{k+i|k} = AP_{k+i|k}C' + R_{vv} \quad (\text{E.18})$$

Hence, the single-step  $j$ -step-ahead prediction error problem may be stated as

$$\mathbf{y}_{k+j}|\mathcal{I}_k = \hat{\mathbf{y}}_{k+j|k}(\theta) + \boldsymbol{\varepsilon}_{k+j,k}|\mathcal{I}_k \quad k = -1, 0, \dots, N-1-j \quad (\text{E.19})$$

with  $\boldsymbol{\varepsilon}_{k+j,k}|\mathcal{I}_k \sim N(0, R_{k+j|k})$  in the ideal case when the system and the model on which the predictor is computed are identical.  $\boldsymbol{\varepsilon}_{k+j,k}$  denotes the residual of the single-step  $j$ -step-ahead predictor at time  $k$ . This corresponds to a standard regression problem in which some measure of the  $j$ -step prediction error

$$\boldsymbol{\varepsilon}_{k+j|k} = \mathbf{y}_{k+j} - \hat{\mathbf{y}}_{k+j|k}(\theta), \quad k = -1, 0, \dots, N-1-j \quad (\text{E.20})$$

is minimized.  $\boldsymbol{\varepsilon}_{k+j|k}$  can be regarded as the realization of  $\boldsymbol{\varepsilon}_{k+j,k}|\mathcal{I}_k \sim N(0, R_{k+j|k})$ . When the structure of the model and the system are different,  $\boldsymbol{\varepsilon}_{k+j,k}|\mathcal{I}_k$  may have a non-zero mean and a covariance different from  $R_{k+j|k}$ . Even in such cases it seems reasonable to minimize some measure of the prediction error,  $\boldsymbol{\varepsilon}_{k+j|k}$ . However, as the distribution of  $\boldsymbol{\varepsilon}_{k+j,k}|\mathcal{I}_k$  is unknown maximum likelihood based procedures can only be considered as approximation, i.e. quasi maximum likelihood.

The least squares  $j$ -step prediction-error estimate

$$\hat{\theta}_{j,LS} = \arg \min_{\theta \in \Theta} \left\{ V_{j,LS}(\theta) = \sum_{k=-1}^{N-1-j} \boldsymbol{\varepsilon}'_{k+j|k} \boldsymbol{\varepsilon}_{k+j|k} \right\} \quad (\text{E.21})$$

is obtained by minimizing the  $j$ -step prediction-error least squares criterion. Similarly, the weighted least squares  $j$ -step prediction error estimate

$$\hat{\theta}_{j,WLS} = \arg \min_{\theta \in \Theta} \left\{ V_{j,WLS}(\theta) = \sum_{k=-1}^{N-1-j} \boldsymbol{\varepsilon}'_{k+j|k} Q \boldsymbol{\varepsilon}_{k+j|k} \right\} \quad (\text{E.22})$$

is obtained by minimizing the  $j$ -step prediction error weighted least squares criterion. The weights,  $Q$ , are selected as the weights in the regulator objective function. The (quasi) maximum likelihood estimate is

$$\hat{\theta}_{j,ML} = \arg \min_{\theta \in \Theta} V_{j,ML}(\theta) \quad (\text{E.23})$$

in which

$$\begin{aligned} V_{j,ML}(\theta) = & \frac{(N+1-j)n_y}{2} \ln(2\pi) \\ & + \frac{1}{2} \sum_{k=-1}^{N-1-j} \ln(\det R_{k+j|k}) \\ & + \frac{1}{2} \sum_{k=-1}^{N-1-j} \boldsymbol{\varepsilon}'_{k+j|k} R_{k+j|k}^{-1} \boldsymbol{\varepsilon}_{k+j|k} \end{aligned} \quad (\text{E.24})$$



is the negative log likelihood function. In its formulation it has been used that  $\varepsilon_{k+j|k} \perp \varepsilon_{l+j|l}$  for  $k \neq l$  (Kailath *et al.*, 2000). The maximum a posteriori estimate is

$$\hat{\theta}_{j,MAP} = \arg \min_{\theta \in \Theta} V_{j,MAP}(\theta) \quad (\text{E.25})$$

in which

$$\begin{aligned} V_{j,MAP}(\theta) &= V_{j,ML}(\theta) \\ &+ \frac{n_\theta}{2} \ln(2\pi) + \frac{1}{2} \ln(\det P_{\theta_0}) \\ &+ \frac{1}{2} (\theta - \theta_0)' P_{\theta_0}^{-1} (\theta - \theta_0) \end{aligned} \quad (\text{E.26})$$

$\theta_0$  is the a priori estimate and  $P_{\theta_0}$  the associated covariance. Other formulations of the  $j$ -step prediction-error problem exist, e.g. regularized  $j$ -step least squares estimation.

As  $\hat{y}_{k+j|k}(\theta)$  is not a simple function of  $\theta$ , the analytical derivatives of  $\varepsilon_{k+j|k} = \varepsilon_{k+j|k}(\theta)$  with respect to  $\theta$  are generally not available. Hence, the optimization algorithms for solving the parameter estimation problem must compute the derivatives of the objective functions numerically, i.e. by finite difference.

The one-step prediction-error estimates may be regarded as special versions of the  $j$ -step prediction-error estimates. However, in that case no extra effort is needed for computing  $\varepsilon_{k+1|k}$  and  $R_{k+1|k}$  as they are already computed as part of the Kalman filter updates. In the  $j$ -step prediction case with  $j > 1$ ,  $\varepsilon_{k+j|k}$  and  $R_{k+j|k}$  must be computed explicitly if needed in the parameter estimation objective function.

Simplifications exist for some class of models, e.g. ARMAX models, for which the covariance matrices are constant as it applies to stochastic stationary process. In addition the gradients can be computed analytically for ARMAX models (Åström, 1980). This simplifies the computation of the parameters significantly.

### E.3.4 Multi-Step Prediction Error

In the multi-step prediction error method, the parameters are estimated such that they minimize the sum of the one- to  $N_p$ -prediction error. The least squares multi-step parameter estimate is

$$\begin{aligned} \hat{\theta}_{1:N_p,LS} &= \\ \arg \min_{\theta \in \Theta} &\left\{ V_{1:N_p,LS}(\theta) = \sum_{j=1}^{N_p} V_{j,LS}(\theta) \right\} \end{aligned} \quad (\text{E.27})$$

and the objective function is efficiently computed as

$$\begin{aligned}
 V_{1:N_p,LS}(\theta) &= \sum_{j=1}^{N_p} V_{j,LS}(\theta) \\
 &= \sum_{j=1}^{N_p} \left( \sum_{k=-1}^{N-1-j} \varepsilon'_{k+j|k} \varepsilon_{k+j|k} \right) \\
 &= \sum_{k=-1}^{N-1-N_p} \sum_{j=1}^{N_p} \varepsilon'_{k+j|k} \varepsilon_{k+j|k} \\
 &\quad + \sum_{k=N-N_p}^{N-2} \sum_{j=1}^{N-1-k} \varepsilon'_{k+j|k} \varepsilon_{k+j|k}
 \end{aligned} \tag{E.28}$$

By this formulation it is evident, that the extra computational expense of computing the multi-step least squares objective function is negligible compared to computation of the  $N_p$ -step least squares objective function.

Similarly, the multi-step weighted least-squares estimate is computed as

$$\begin{aligned}
 \hat{\theta}_{1:N_p,WLS} &= \\
 \arg \min_{\theta \in \Theta} &\left\{ V_{1:N_p,WLS}(\theta) = \sum_{j=1}^{N_p} V_{j,WLS}(\theta) \right\}
 \end{aligned} \tag{E.29}$$

in which the objective function is evaluated as

$$\begin{aligned}
 V_{1:N_p,WLS}(\theta) &= \sum_{k=-1}^{N-1-N_p} \sum_{j=1}^{N_p} \varepsilon'_{k+j|k} Q \varepsilon_{k+j|k} \\
 &\quad + \sum_{k=N-N_p}^{N-2} \sum_{j=1}^{N-1-k} \varepsilon'_{k+j|k} Q \varepsilon_{k+j|k}
 \end{aligned} \tag{E.30}$$

The multi-step pseudo maximum-likelihood prediction error estimate may be defined similarly

$$\begin{aligned}
 \hat{\theta}_{1:N_p,pML} &= \\
 \arg \min_{\theta \in \Theta} &\left\{ V_{1:N_p,pML}(\theta) = \sum_{j=1}^{N_p} V_{j,pML}(\theta) \right\}
 \end{aligned} \tag{E.31}$$

with the objective function computed as

$$\begin{aligned}
 V_{1:N_p, pML}(\theta) &= \sum_{j=1}^{N_p} \sum_{k=-1}^{N-1-j} \frac{N+1-j}{2} \ln(2\pi) + \\
 &\frac{1}{2} \sum_{k=-1}^{N-1-N_p} \sum_{j=1}^{N_p} \left[ \ln(\det R_{k+j|k}) + \varepsilon'_{k+j|k} R_{k+j|k}^{-1} \varepsilon_{k+j|k} \right] + \\
 &\frac{1}{2} \sum_{k=N-N_p}^{N-2} \sum_{j=1}^{N-1-k} \left[ \ln(\det R_{k+j|k}) + \varepsilon'_{k+j|k} R_{k+j|k}^{-1} \varepsilon_{k+j|k} \right]
 \end{aligned} \tag{E.32}$$

Similarly, the multi-step pseudo maximum a posteriori estimate is computed as

$$\begin{aligned}
 \hat{\theta}_{1:N_p, pMAP} &= \\
 \arg \min_{\theta \in \Theta} &\left\{ V_{1:N_p, pMAP}(\theta) = \sum_{j=1}^{N_p} V_{j, MAP}(\theta) \right\}
 \end{aligned} \tag{E.33}$$

in which

$$\begin{aligned}
 V_{1:N_p, pMAP}(\theta) &= V_{1:N_p, pML}(\theta) + \frac{n_\theta}{2} \ln(2\pi) \\
 &+ \frac{1}{2} \ln(\det P_{\theta_0}) + \frac{1}{2} (\theta - \theta_0)' P_{\theta_0}^{-1} (\theta - \theta_0)
 \end{aligned} \tag{E.34}$$

The above multi-step maximum likelihood and maximum a posteriori estimates are called pseudo maximum likelihood and pseudo maximum a posteriori, respectively, because the correlations between  $\varepsilon_{k+i,k}|\mathcal{I}_k$  and  $\varepsilon_{k+j,k}|\mathcal{I}_k$  for  $i \neq j$  are neglected. The formulas for these estimates have been introduced heuristically by summing corresponding  $j$ -step prediction-error objective functions as could be done in the least-squares and weighted least-squares case. No statistical considerations are taken into account in deriving these estimates.

### E.3.5 Multi-Step Maximum Likelihood Predictors

To deduce true multi-step prediction-error (quasi) maximum likelihood and maximum a posteriori estimators, the correlation between  $\varepsilon_{k+i,k}|\mathcal{I}_k$  and  $\varepsilon_{k+j,k}|\mathcal{I}_k$  for  $i \neq j$  must be taken into account. This correlation is

$$\langle \varepsilon_{k+i,k}|\mathcal{I}_k, \varepsilon_{k+j,k}|\mathcal{I}_k \rangle = R_{(i,j)|k} \tag{E.35}$$

Define

$$\begin{aligned}
 \mathbf{Y}_k &= [\mathbf{y}'_{k+1} \quad \dots \quad \mathbf{y}'_{k+N_p}]' \quad k = -1, 0, \dots, N-1-N_p \\
 \mathbf{Y}_k &= [\mathbf{y}'_{k+1} \quad \dots \quad \mathbf{y}'_{N-1}]' \quad k = N-N_p, \dots, N-2
 \end{aligned}$$

and the corresponding multi-step predictions

$$\begin{aligned}\hat{Y}_k(\theta) &= \begin{bmatrix} \hat{y}'_{k+1|k} & \cdots & \hat{y}'_{k+N_p|k} \end{bmatrix}' \quad k = -1, 0, \dots, N-1-N_p \\ \hat{Y}_k(\theta) &= \begin{bmatrix} \hat{y}'_{k+1|k} & \cdots & \hat{y}'_{N-1|k} \end{bmatrix}' \quad k = N-N_p, \dots, N-2\end{aligned}$$

Furthermore, define the conditional multi-step prediction error vector as

$$\epsilon_k | \mathcal{I}_k = \begin{bmatrix} \epsilon_{k+1,k} | \mathcal{I}_k \\ \epsilon_{k+2,k} | \mathcal{I}_k \\ \vdots \\ \epsilon_{k+N_p,k} | \mathcal{I}_k \end{bmatrix} \quad \epsilon_k | \mathcal{I}_k = \begin{bmatrix} \epsilon_{k+1,k} | \mathcal{I}_k \\ \epsilon_{k+2,k} | \mathcal{I}_k \\ \vdots \\ \epsilon_{N-1,k} | \mathcal{I}_k \end{bmatrix} \quad (\text{E.37})$$

for  $k = -1, 0, \dots, N-1-N_p$  (the left vector) and  $k = N-N_p, \dots, N-2$  (the right vector), respectively.

The multi-step prediction error problem can then be expressed as the stochastic model

$$\mathbf{Y}_k | \mathcal{I}_k = \hat{Y}_k(\theta) + \epsilon_k | \mathcal{I}_k \quad k = -1, 0, \dots, N-2 \quad (\text{E.38})$$

with  $\epsilon_k | \mathcal{I}_k \sim N(0, R_k)$ . For  $k = -1, 0, \dots, N-1-N_p$ ,  $R_k$  is

$$\begin{aligned}R_k &= \langle \epsilon_k | \mathcal{I}_k, \epsilon_k | \mathcal{I}_k \rangle \\ &= \begin{bmatrix} R_{(1,1)|k} & R_{(1,2)|k} & \cdots & R_{(1,N_p)|k} \\ R_{(2,1)|k} & R_{(2,2)|k} & \cdots & R_{(2,N_p)|k} \\ \vdots & \vdots & & \vdots \\ R_{(N_p,1)|k} & R_{(N_p,2)|k} & \cdots & R_{(N_p,N_p)|k} \end{bmatrix} \quad (\text{E.39})\end{aligned}$$

For  $k = N-N_p, \dots, N-2$ ,  $R_k$  is

$$\begin{aligned}R_k &= \langle \epsilon_k | \mathcal{I}_k, \epsilon_k | \mathcal{I}_k \rangle \\ &= \begin{bmatrix} R_{(1,1)|k} & R_{(1,2)|k} & \cdots & R_{(1,N-1-k)|k} \\ R_{(2,1)|k} & R_{(2,2)|k} & \cdots & R_{(2,N-1-k)|k} \\ \vdots & \vdots & & \vdots \\ R_{(N-1-k,1)|k} & R_{(N-1-k,2)|k} & \cdots & R_{(N-1-k,N-1-k)|k} \end{bmatrix} \quad (\text{E.40})\end{aligned}$$

The realization of the multi-step prediction-error vector for  $k = -1, 0, \dots, N-1-N_p$  is

$$\begin{aligned}\epsilon_{k|k} &= Y_k - \hat{Y}_k(\theta) \\ &= \begin{bmatrix} y_{k+1} - \hat{y}_{k+1|k} \\ y_{k+2} - \hat{y}_{k+2|k} \\ \vdots \\ y_{k+N_p} - \hat{y}_{k+N_p|k} \end{bmatrix} = \begin{bmatrix} \epsilon_{k+1|k} \\ \epsilon_{k+2|k} \\ \vdots \\ \epsilon_{k+N_p|k} \end{bmatrix} \quad (\text{E.41})\end{aligned}$$

and

$$\begin{aligned}\epsilon_{k|k} &= Y_k - \hat{Y}_k(\theta) \\ &= \begin{bmatrix} y_{k+1} - \hat{y}_{k+1|k} \\ y_{k+2} - \hat{y}_{k+2|k} \\ \vdots \\ y_{N-1} - \hat{y}_{N-1|k} \end{bmatrix} = \begin{bmatrix} \varepsilon_{k+1|k} \\ \varepsilon_{k+2|k} \\ \vdots \\ \varepsilon_{N-1|k} \end{bmatrix}\end{aligned}\quad (\text{E.42})$$

for  $k = N - N_p, \dots, N - 2$ .

The negative log likelihood function for the multi-step prediction is

$$\begin{aligned}V_{1:N_p, ML}(\theta) &= \sum_{k=-1}^{N-1-N_p} \frac{n_y N_p}{2} \ln(2\pi) \\ &\quad + \sum_{k=N-N_p}^{N-2} \frac{n_y(N-1-k)}{2} \ln(2\pi) \\ &\quad + \frac{1}{2} \sum_{k=-1}^{N-2} \left( \ln(\det R_k) + \epsilon'_{k|k} R_k^{-1} \epsilon_{k|k} \right) \\ &= \frac{n_y f}{2} \ln(2\pi) \\ &\quad + \frac{1}{2} \sum_{k=-1}^{N-2} \left( \ln(\det R_k) + \epsilon'_{k|k} R_k^{-1} \epsilon_{k|k} \right)\end{aligned}\quad (\text{E.43})$$

in which  $f = N_p \left[ N - \frac{1}{2}(N_p - 1) \right]$ . Consequently, the multi-step prediction error (quasi) maximum likelihood estimate is

$$\hat{\theta}_{1:N_p, ML} = \arg \min_{\theta \in \Theta} V_{1:N_p, ML}(\theta) \quad (\text{E.44})$$

Similarly, the multi-step prediction error (quasi) maximum a posteriori estimate is

$$\hat{\theta}_{1:N_p, MAP} = \arg \min_{\theta \in \Theta} V_{1:N_p, MAP}(\theta) \quad (\text{E.45})$$

in which

$$\begin{aligned}V_{1:N_p, MAP}(\theta) &= V_{1:N_p, ML}(\theta) + \frac{n_\theta}{2} \ln(2\pi) \\ &\quad + \frac{1}{2} \ln(\det P_{\theta_0}) + \frac{1}{2} (\theta - \theta_0)' P_{\theta_0}^{-1} (\theta - \theta_0)\end{aligned}\quad (\text{E.46})$$

In computation of the multi-step prediction-error maximum likelihood estimate,  $\ln(\det R_k)$  and  $\epsilon'_{k|k} R_k^{-1} \epsilon_{k|k}$  must be computed.  $\epsilon_{k|k}$  is obtained by computing the  $j$ -step prediction errors. This is accomplished using (E.7) for  $j = 1, 2, \dots, N_p$  given  $\mathcal{I}_k$  and  $\hat{u}_{k+j|k} = u_{k+j}$ . By construction the covariance

matrix,  $R_k$ , has the special structure that arise from a state space model. This implies that (Kailath *et al.*, 2000)

$$R_k = L_k R_{\epsilon,k} L_k' \quad (\text{E.47})$$

in which the factorization,  $L_k$  and  $R_{\epsilon,k}$ , is computed using the Kalman filter recursions (E.2c)-(E.2e), (E.4) and (E.6a). Using the one-step predictive Kalman gain

$$K_{p,k} = AK_{fx,k} + K_{fw,k} \quad (\text{E.48})$$

the block lower triangular matrix,  $L_k$ , may be computed as

$$L_k = \begin{bmatrix} I & 0 & \dots & 0 \\ CK_{p,k+1} & I & \dots & 0 \\ CAK_{p,k+1} & CK_{p,k+2} & \dots & 0 \\ \vdots & \vdots & & \vdots \\ CA^{N_p-2}K_{p,k+1} & CA^{N_p-3}K_{p,k+2} & \dots & I \end{bmatrix} \quad (\text{E.49})$$

and the block diagonal matrix,  $R_{\epsilon,k}$ , is

$$R_{\epsilon,k} = \begin{bmatrix} R_{e,k+1} & & & \\ & R_{e,k+2} & & \\ & & \ddots & \\ & & & R_{e,k+N_p} \end{bmatrix} \quad (\text{E.50})$$

Hence, the determinant of  $R_k$  may be computed as

$$\det R_k = \det R_{\epsilon,k} = \prod_{j=1}^{N_p} \det R_{e,k+j} \quad (\text{E.51})$$

which implies

$$\ln(\det R_k) = \ln \left( \prod_{j=1}^{N_p} \det R_{e,k+j} \right) = \sum_{j=1}^{N_p} \ln(\det R_{e,k+j}) \quad (\text{E.52})$$

Consequently, the term  $\sum_{k=-1}^{N-2} \ln(\det R_k)$  in (E.43) may be evaluated as

$$\begin{aligned} \sum_{k=-1}^{N-2} \ln(\det R_k) &= \sum_{k=0}^{N_p-2} (k+1) \ln(\det R_{e,k}) \\ &\quad + N_p \sum_{k=N_p-1}^{N-1} \ln(\det R_{e,k}) \end{aligned} \quad (\text{E.53})$$

The term  $\epsilon'_{k|k} R_k^{-1} \epsilon_{k|k}$  can be evaluated as

$$\begin{aligned} \epsilon'_{k|k} R_k^{-1} \epsilon_{k|k} &= \epsilon'_{k|k} (L_k R_{e,k} L_k')^{-1} \epsilon_{k|k} \\ &= (L_k^{-1} \epsilon_{k|k})' R_{e,k}^{-1} (L_k^{-1} \epsilon_{k|k}) \\ &= \sum_{j=1}^{N_p} \bar{e}'_{k+j|k} R_{e,k+j}^{-1} \bar{e}_{k+j|k} \end{aligned} \quad (\text{E.54})$$

in which  $\begin{bmatrix} \bar{e}'_{k+1|k} & \bar{e}'_{k+2|k} & \cdots & \bar{e}'_{k+N_p|k} \end{bmatrix}' = L_k^{-1} \epsilon_{k|k} \cdot \{\bar{e}_{k+j|k}\}_{j=1}^{N_p}$  is efficiently computed using the Kalman filter recursions for  $j = 1, 2, \dots, N_p$

$$\bar{e}_{k+j|k} = \varepsilon_{k+j|k} - C \bar{x}_{k+j|k} \quad (\text{E.55a})$$

$$\bar{x}_f = \bar{x}_{k+j|k} + K_{fx,k+j} \bar{e}_{k+j|k} \quad (\text{E.55b})$$

$$\bar{w}_f = K_{fw,k+j} \bar{e}_{k+j|k} \quad (\text{E.55c})$$

$$\bar{x}_{k+j+1|k} = A \bar{x}_f + \bar{w}_f \quad (\text{E.55d})$$

with  $\bar{x}_{k+1|k} = 0$ . Note that (E.55b)-(E.55d) may be expressed as

$$\bar{x}_{k+j+1|k} = A \bar{x}_{k+j|k} + K_{p,k+j} \bar{e}_{k+j|k} \quad (\text{E.56})$$

which implies that (E.55) can be expressed as

$$\bar{x}_{k+j+1|k} = (A - K_{p,k+j} C) \bar{x}_{k+j|k} + K_{p,k+j} \varepsilon_{k+j|k} \quad (\text{E.57a})$$

$$\bar{e}_{k+j|k} = -C \bar{x}_{k+j|k} + \varepsilon_{k+j|k} \quad (\text{E.57b})$$

Consequently, the term  $\sum_{k=-1}^{N-2} \epsilon'_{k|k} R_k^{-1} \epsilon_{k|k}$  in (E.43) may be efficiently evaluated using

$$\begin{aligned} \sum_{k=-1}^{N-2} \epsilon'_{k|k} R_k^{-1} \epsilon_{k|k} &= \sum_{k=0}^{N_p-2} \sum_{j=1}^{k+1} \bar{e}'_{k|k-j} R_{e,k}^{-1} \bar{e}_{k|k-j} \\ &\quad + \sum_{k=N_p-1}^{N-1} \sum_{j=1}^{N_p} \bar{e}'_{k|k-j} R_{e,k}^{-1} \bar{e}_{k|k-j} \end{aligned} \quad (\text{E.58})$$

and a bank of Kalman filter recursion (E.55) for computing  $\bar{e}_{k|k-j}$  and  $\bar{x}_{k+1|k-j}$  for  $j = 1, 2, \dots, N_p$ . Hence, at each time instant  $k$  the multi-step prediction error  $\epsilon_{k|k}$  is computed using the Kalman predictions. This vector is stored in memory for  $N_p$  iterations such that  $\varepsilon_{k|k-j}$  can be used in computation of  $\bar{e}_{k|k-j}$  and subsequent evaluation of the terms in (E.58). The advantage of this method compared to a naive implementation is that gains and covariances in the Kalman recursions need to be evaluated only once at each time step.

## E.4 Continuous-Time Stochastic Model

System models based on conservation laws and formulated using differential balances are most conveniently modelled in the continuous time domain. Furthermore, continuous-time models often contain much fewer parameters than equivalent discrete-time models. Hence, instead of proposing a system modelled by stochastic linear difference equations (E.1), a system modelled by stochastic linear differential equations is proposed (Jazwinski, 1970; Åström, 1970a, 1980), i.e.

$$d\mathbf{x}(t) = [F(\theta)\mathbf{x}(t) + G(\theta)u(t)] dt + H(\theta)d\boldsymbol{\omega}(t) \quad (\text{E.59a})$$

$$\mathbf{y}(t_k) = C(\theta)\mathbf{x}(t_k) + \mathbf{v}(t_k) \quad (\text{E.59b})$$

in which  $\{\boldsymbol{\omega}(t)\}$  is a standard Wiener process and  $\mathbf{v}(t_k) \sim N(0, R_{vv}(\theta))$ . The stochastic differential equation (E.59a) contains a drift term,  $(F\mathbf{x}(t) + Gu(t))dt$ , as well as a diffusion term,  $Hd\boldsymbol{\omega}(t)$ . The structure and parameter dependence of the drift term may be obtained in the usual way by forming differential balances and linearization around a steady state. The diffusion term is used to represent unknown disturbances and plant-model mismatch. In the control literature, the diffusion term is known as the disturbance model.

The continuous-discrete time system (E.59) may be represented as an equivalent discrete-time system (E.1). The output equation (E.59b) of the continuous-discrete time system is identical to the output equation (E.1b) of the discrete-time system. Hence the matrix,  $C$ , and the covariance matrix,  $R_{vv}$ , in the discrete time system (E.1) are identical to their counterparts in the continuous-discrete time system (E.59).  $R_{ww} = 0$  in the discrete-time model (E.1) as  $\mathbf{v}(t_k)$  and the standard Wiener process  $\{\boldsymbol{\omega}(t)\}$  are independent in the continuous-discrete time model (E.59). The stochastic differential equation (E.59a) is related to the stochastic difference equation (E.1a) through the expressions (Åström, 1970a; Åström and Wittenmark, 1997)

$$A = A(T_s) = \exp(FT_s) \quad (\text{E.60a})$$

$$B = B(T_s) = \int_0^{T_s} \exp(Fs) ds G \quad (\text{E.60b})$$

$$R_{ww} = R_{ww}(T_s) = \int_0^{T_s} \exp(Fs) HH' \exp(F's) ds \quad (\text{E.60c})$$

in which  $T_s$  is the sampling time.  $A$ ,  $B$ , and  $R_{ww}$  are efficiently computed using the matrix exponential relation

$$\exp\left(\begin{bmatrix} -F & HH' & 0 \\ 0 & F' & I \\ 0 & 0 & 0 \end{bmatrix} t\right) = \begin{bmatrix} F_1(t) & G_1(t) & H_1(t) \\ 0 & F_2(t) & G_2(t) \\ 0 & 0 & F_3(t) \end{bmatrix} \quad (\text{E.61})$$



and the expressions (Moler and Van Loan, 1978; Van Loan, 1978; Sidje, 1998)

$$A = A(T_s) = F_2(T_s)' \quad (\text{E.62a})$$

$$B = B(T_s) = G_2(T_s)'G \quad (\text{E.62b})$$

$$R_{ww} = R_{ww}(T_s) = F_2(T_s)'G_1(T_s) \quad (\text{E.62c})$$

For numerical implementation, we find the discretization approach discussed here more instructive and efficient than the alternative procedure that establishes differential equations for the conditional covariance evolution (Jazwinski, 1970; Åström, 1980; Kristensen *et al.*, 2004c). In the direct approach described here, the conditional covariance evolution,  $P_{k+1|k}$ , is computed through the state sensitivities,  $A$ , and (E.6a) rather than through solution of a differential equation for the covariance evolution, i.e. a differential equation of the form

$$\frac{d}{dt}P(t|t_k) = FP(t|t_k) + P(t|t_k)F' + HH' \quad (\text{E.63})$$

In the linear case, the state sensitivities are efficiently computed using (E.62a). Even in the nonlinear case, it is computationally more efficient to compute the conditional covariance evolution by computation of the state sensitivities,  $A$ , and (E.6a) rather than by an expression similar to (E.63). The reason for this is existence of very efficient methods for joint integration and state sensitivity computation (Kristensen *et al.*, 2004a).

While systems of stochastic differential equations (E.59) are useful for representing dynamics with a stochastic component, they suffer from the shortcoming that they are not suitable for modelling distributed systems. Most industrial processes are distributed. One way to represent multivariate stochastic distributed processes is through the input-output representation in the LaPlace domain

$$\mathbf{Z}(s) = G(s; \theta)U(s) + H(s; \theta)\mathbf{E}(s) \quad (\text{E.64a})$$

$$\mathbf{y}(t_k) = \mathbf{z}(t_k) + \mathbf{v}(t_k) \quad (\text{E.64b})$$

in which  $U(s)$  is the process input vector,  $\mathbf{E}(s)$  is a vector with white noise components,  $\mathbf{Z}(s)$  is the process output vector.  $\mathbf{v}(t_k)$  is the measurement noise vector and  $\mathbf{y}(t_k)$  is the measured process output vector at time  $t_k$ . The elements,  $\{g_{ij}(s)\}$  and  $\{h_{ij}(s)\}$ , of the transfer function matrices,  $G(s)$  and  $H(s)$ , are rational transfer functions with time delays

$$g_{ij}(s) = \frac{b_{ij}(s; \theta)}{a_{ij}(s; \theta)} \exp(-\tau_{ij}(\theta)s) \quad (\text{E.65a})$$

$$h_{ij}(s) = \frac{d_{ij}(s; \theta)}{c_{ij}(s; \theta)} \exp(-\lambda_{ij}(\theta)s) \quad (\text{E.65b})$$

In the case of no delays, the transfer function representation (E.64) can be realized as a system of stochastic differential equations (E.59) and transformed

to an equivalent discrete-time representation (E.1). In the case with delays, the transfer function is formally realized as a system of delayed stochastic differential equations and then converted to an equivalent discrete-time representation (E.1). The Kalman filter and predictor is then used as the optimal estimator for (E.1) and used to compute the various prediction-error criteria for the parameter estimation. This procedure corresponds to computation of the optimal predictions for the system (E.64) described using the transfer functions (E.65).

## E.5 SISO Example

To illustrate the identification criteria discussed in this paper, we consider the SISO system,  $\mathcal{S} = \{g(s), h(s)\}$ , defined as

$$\mathbf{Z}(s) = g(s)U(s) + h(s)\mathbf{E}(s) \quad (\text{E.66a})$$

$$\mathbf{y}(t_k) = \mathbf{z}(t_k) + \mathbf{v}(t_k) \quad (\text{E.66b})$$

in which  $\mathbf{E}(s)$  is standard white noise and  $\mathbf{v}(t_k) \sim N_{iid}(0, r^2)$ . The transfer function,  $g(s)$ , from the process inputs,  $U(s)$ , to the process output,  $\mathbf{Y}(s)$ , and the disturbance transfer function,  $h(s)$ , are

$$g(s) = \frac{K}{(\alpha_1 s + 1)(\alpha_2 s + 1)} e^{-\tau s} \quad (\text{E.67a})$$

$$h(s) = \frac{\sigma}{\gamma s + 1} \quad (\text{E.67b})$$

The parameters defining the system  $\mathcal{S}$  and used for generating the data are:  $K = 1.0$ ,  $\alpha_1 = 1.0$ ,  $\alpha_2 = 3.0$ ,  $\tau = 5.2$ ,  $\sigma = 0.2$ ,  $\gamma = 1.0$  and  $r = 0.2$ . The system is sampled with a sampling time of  $T_s = 0.25$ . The deterministic input,  $U(s)$ , is assumed to be implemented using a zero-order-hold circuit. The IO-data used for estimation of this system are illustrated in figure E.1.

### E.5.1 Identical Model and System Structure

Consider the situation in which the model and the system has the same structure. In this case the structure of the model,  $\hat{g}(s)$ , and the disturbance model,  $\hat{h}(s)$ , are

$$\hat{g}(s) = \frac{\hat{K}}{(\hat{\alpha}_1 s + 1)(\hat{\alpha}_2 s + 1)} e^{-\hat{\tau} s} \quad (\text{E.68a})$$

$$\hat{h}(s) = \frac{\hat{\sigma}}{\hat{\gamma} s + 1} \quad (\text{E.68b})$$

Let  $\mathcal{M} = \{\hat{g}(s), \hat{h}(s)\}$ . This implies that the true system,  $\mathcal{S}$ , is within the class of models,  $\mathcal{M}$ , estimated, i.e.  $\mathcal{S} \in \mathcal{M}$ .

**Table E.1.** Single-Step LS Estimation.  $\theta_0 = \theta$ . Model (E.68). For  $j = 40$ , apparently the optimizer converges to a local minimum.

| $j$ | $K$    | $\alpha_1$ | $\alpha_2$ | $\tau$ | $\sigma$ | $\gamma$ | $r$                  | $\sigma/r$        | $V$   | CPU sec. |
|-----|--------|------------|------------|--------|----------|----------|----------------------|-------------------|-------|----------|
| 1   | 0.9797 | 0.5643     | 3.4213     | 5.3171 | 0.3066   | 1.3754   | 0.2887               | 1.0618            | 109.9 | 75       |
| 4   | 0.9792 | 0.5799     | 3.4052     | 5.3118 | 0.5135   | 1.1552   | 0.4016               | 1.2787            | 116.9 | 83       |
| 8   | 0.9790 | 0.7231     | 3.3499     | 5.2044 | 0.4699   | 1.1888   | 0.3192               | 1.4722            | 120.0 | 73       |
| 20  | 0.9832 | 0.7091     | 3.3811     | 5.2017 | 3.2526   | 9.7835   | 3.4802               | 0.9346            | 119.2 | 188      |
| 40  | 0.9789 | 0.8811     | 3.2721     | 5.1013 | 1.1390   | 3.4448   | $2.12 \cdot 10^{-5}$ | $5.37 \cdot 10^4$ | 122.2 | 185      |
| 80  | 0.9719 | 0.7614     | 3.3373     | 5.1576 | 0.2000   | 1.0000   | 0.2000               | 1.0000            | 129.5 | 196      |
| 100 | 1.0087 | 0.9819     | 3.3473     | 4.8655 | 0.2000   | 1.0000   | 0.2000               | 1.0000            | 200.3 | 194      |
| 200 | 0.9426 | 1.1232     | 2.8932     | 5.0718 | 0.2000   | 1.0000   | 0.2000               | 1.0000            | 130.4 | 373      |

**Table E.2.** Single-Step LS Estimation.  $\theta_0 = 0.9 \cdot \theta$ . Model (E.68).

| $j$ | $K$    | $\alpha_1$ | $\alpha_2$ | $\tau$ | $\sigma$ | $\gamma$ | $r$    | $\sigma/r$ | $V$   | CPU sec. |
|-----|--------|------------|------------|--------|----------|----------|--------|------------|-------|----------|
| 1   | 0.9797 | 0.5641     | 3.4216     | 5.3171 | 0.8705   | 1.3757   | 0.8198 | 1.0618     | 109.9 | 113      |
| 4   | 0.9792 | 0.5798     | 3.4053     | 5.3119 | 0.4802   | 1.1548   | 0.3756 | 1.2785     | 116.9 | 154      |
| 8   | 0.9790 | 0.7239     | 3.3496     | 5.2037 | 0.8301   | 1.1885   | 0.5636 | 1.4730     | 120.0 | 227      |
| 20  | 0.9832 | 0.7086     | 3.3815     | 5.2019 | 1.9202   | 9.7771   | 2.0543 | 0.9347     | 119.2 | 351      |
| 40  | 0.9786 | 0.8639     | 3.2871     | 5.1016 | 0.1824   | 0.9056   | 0.1776 | 1.0268     | 122.5 | 325      |
| 80  | 0.9719 | 0.7612     | 3.3374     | 5.1578 | 0.1800   | 0.9000   | 0.1800 | 1.0000     | 129.5 | 394      |
| 100 | 1.0087 | 0.9820     | 3.3471     | 4.8656 | 0.1800   | 0.9000   | 0.1800 | 1.0000     | 200.3 | 376      |
| 200 | 0.9428 | 1.1445     | 2.8801     | 5.0634 | 0.1800   | 0.9000   | 0.1800 | 1.0000     | 130.4 | 532      |

**Table E.3.** Multi-Step LS Estimation.  $\theta_0 = \theta$ . Model (E.68).

| $N_p$ | $K$    | $\alpha_1$ | $\alpha_2$ | $\tau$ | $\sigma$ | $\gamma$ | $r$    | $\sigma/r$ | $V$   | CPU sec. |
|-------|--------|------------|------------|--------|----------|----------|--------|------------|-------|----------|
| 1     | 0.9797 | 0.5652     | 3.4209     | 5.3164 | 0.3082   | 1.3750   | 0.2902 | 1.0618     | 110.0 | 64       |
| 4     | 0.9796 | 0.5659     | 3.4194     | 5.3169 | 0.2347   | 1.3862   | 0.2214 | 1.0603     | 449.1 | 120      |
| 8     | 0.9794 | 0.6137     | 3.3979     | 5.2827 | 0.5525   | 1.3641   | 0.5220 | 1.0585     | 924.2 | 175      |
| 20    | 0.9792 | 0.7116     | 3.3562     | 5.2107 | 0.2986   | 1.4301   | 0.2833 | 1.0540     | 2370  | 276      |
| 40    | 0.9823 | 0.7394     | 3.3641     | 5.1836 | 0.0277   | 8.0822   | 0.0295 | 0.9382     | 4763  | 804      |
| 80    | 0.9804 | 0.7596     | 3.3357     | 5.1767 | 0.2408   | 7.1234   | 0.2530 | 0.9514     | 9481  | 1154     |
| 100   | 0.9796 | 0.7587     | 3.3305     | 5.1782 | 0.4931   | 6.5668   | 0.5253 | 0.9386     | 11804 | 1389     |
| 200   | 0.9759 | 0.7740     | 3.2966     | 5.1802 | 0.9572   | 6.1967   | 1.0277 | 0.9313     | 23023 | 2635     |

**Table E.4.** Multi-Step LS Estimation.  $\theta_0 = 0.9 \cdot \theta$ . Model (E.68).

| $N_p$ | $K$    | $\alpha_1$ | $\alpha_2$ | $\tau$ | $\sigma$ | $\gamma$ | $r$    | $\sigma/r$ | $V$   | CPU sec. |
|-------|--------|------------|------------|--------|----------|----------|--------|------------|-------|----------|
| 1     | 0.9797 | 0.5632     | 3.4219     | 5.3179 | 0.3377   | 1.3754   | 0.3180 | 1.0620     | 110.0 | 87       |
| 4     | 0.9796 | 0.5657     | 3.4194     | 5.3170 | 0.0080   | 1.3861   | 0.0075 | 1.0603     | 449.1 | 421      |
| 8     | 0.9794 | 0.6136     | 3.3981     | 5.2827 | 0.3805   | 1.3641   | 0.3595 | 1.0585     | 924.2 | 251      |
| 20    | 0.9792 | 0.7116     | 3.3563     | 5.2107 | 0.3053   | 1.4301   | 0.2897 | 1.0539     | 2370  | 393      |
| 40    | 0.9823 | 0.7394     | 3.3641     | 5.1836 | 0.7805   | 8.0826   | 0.8318 | 0.9383     | 4763  | 644      |
| 80    | 0.9804 | 0.7597     | 3.3357     | 5.1767 | 0.4734   | 7.1230   | 0.4975 | 0.9514     | 9481  | 1101     |
| 100   | 0.9796 | 0.7586     | 3.3305     | 5.1782 | 0.6825   | 6.5664   | 0.7271 | 0.9386     | 11804 | 1426     |
| 200   | 0.9760 | 0.7739     | 3.2966     | 5.1802 | 0.5728   | 6.1969   | 0.6151 | 0.9314     | 23023 | 2382     |

**Table E.5.** Single-Step ML Estimation.  $\theta_0 = \theta$ . Model (E.68).

| $j$ | $K$    | $\alpha_1$ | $\alpha_2$ | $\tau$ | $\sigma$ | $\gamma$ | $r$    | $\sigma/r$ | $V$    | CPU sec. |
|-----|--------|------------|------------|--------|----------|----------|--------|------------|--------|----------|
| 1   | 0.9797 | 0.5651     | 3.4210     | 5.3164 | 0.2204   | 1.3762   | 0.2077 | 1.0613     | -63.27 | 63       |
| 4   | 0.9792 | 0.5799     | 3.4047     | 5.3121 | 0.2432   | 1.1319   | 0.1868 | 1.3024     | -1.987 | 178      |
| 8   | 0.9789 | 0.7184     | 3.3500     | 5.2089 | 0.2853   | 1.0881   | 0.1526 | 1.8698     | 23.57  | 387      |
| 20  | 0.9833 | 0.7078     | 3.3827     | 5.2018 | 0.2272   | 9.6807   | 0.2394 | 0.9490     | 17.60  | 882      |
| 40  | 0.9786 | 0.8638     | 3.2872     | 5.1061 | 0.0002   | 0.0046   | 0.2475 | 0.0008     | 45.37  | 1288     |
| 80  | 0.9719 | 0.7619     | 3.3372     | 5.1572 | 0.0002   | 0.0164   | 0.2545 | 0.0008     | 100.7  | 3213     |
| 100 | 1.0087 | 0.9817     | 3.3474     | 4.8657 | 0.0002   | 0.0173   | 0.3165 | 0.0006     | 536.9  | 7720     |
| 200 | 0.9426 | 1.1231     | 2.8932     | 5.0718 | 0.0002   | 0.1342   | 0.2553 | 0.0008     | 107.4  | 5043     |

**Table E.6.** Single-Step ML Estimation.  $\theta_0 = 0.9 \cdot \theta$ . Model (E.68).

| $j$ | $K$    | $\alpha_1$ | $\alpha_2$ | $\tau$ | $\sigma$ | $\gamma$ | $r$    | $\sigma/r$ | $V$    | CPU sec. |
|-----|--------|------------|------------|--------|----------|----------|--------|------------|--------|----------|
| 1   | 0.9797 | 0.5651     | 3.4211     | 5.3164 | 0.2204   | 1.3762   | 0.2077 | 1.0613     | -63.27 | 115      |
| 4   | 0.9793 | 0.5798     | 3.4048     | 5.3120 | 0.2432   | 1.1321   | 0.1868 | 1.3022     | -1.987 | 371      |
| 8   | 0.9789 | 0.7184     | 3.3500     | 5.2088 | 0.2853   | 1.0881   | 0.1526 | 1.8694     | 23.57  | 725      |
| 20  | 0.9832 | 0.7081     | 3.3829     | 5.2007 | 0.2243   | 9.0000   | 0.2391 | 0.9384     | 17.62  | 2038     |
| 40  | 0.9786 | 0.8639     | 3.2871     | 5.1060 | 0.0002   | 0.1380   | 0.2475 | 0.0007     | 45.37  | 3297     |
| 80  | 0.9719 | 0.7608     | 3.3376     | 5.1580 | 0.0002   | 0.1985   | 0.2545 | 0.0007     | 100.7  | 6082     |
| 100 | 1.0088 | 0.9817     | 3.3474     | 4.8656 | 0.0002   | 0.2748   | 0.3165 | 0.0006     | 536.9  | 6696     |
| 200 | 0.9426 | 1.1232     | 2.8931     | 5.0718 | 0.0002   | 0.0291   | 0.2553 | 0.0007     | 107.4  | 13227    |

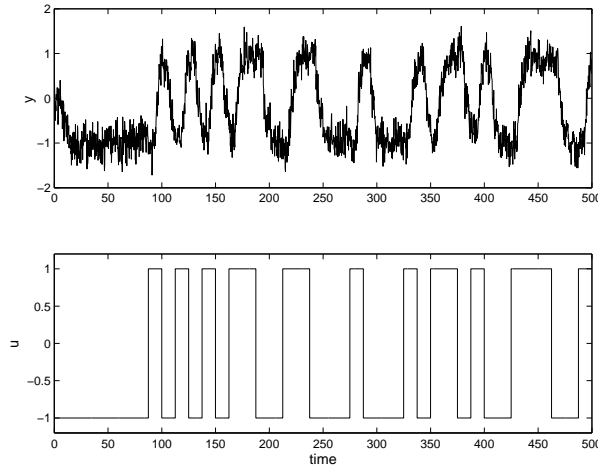


**Table E.7.** Multi-Step ML Estimation.  $\theta_0 = \theta$ . Model (E.68).

| $j$ | $K$    | $\alpha_1$ | $\alpha_2$ | $\tau$ | $\sigma$ | $\gamma$ | $r$    | $\sigma/r$ | $V$    | CPU sec. |
|-----|--------|------------|------------|--------|----------|----------|--------|------------|--------|----------|
| 1   | 0.9797 | 0.5651     | 3.4210     | 5.3164 | 0.2204   | 1.3761   | 0.2077 | 1.0613     | -63.27 | 60       |
| 4   | 0.9798 | 0.5607     | 3.4252     | 5.3186 | 0.2182   | 1.4747   | 0.2102 | 1.0384     | -248.7 | 106      |
| 8   | 0.9798 | 0.5307     | 3.4370     | 5.3425 | 0.2051   | 1.5454   | 0.2140 | 0.9582     | -468.5 | 172      |
| 20  | 0.9801 | 0.5262     | 3.4700     | 5.3392 | 0.1860   | 1.5199   | 0.2189 | 0.8498     | -829.0 | 402      |
| 40  | 0.9835 | 0.5302     | 3.5242     | 5.3124 | 0.1807   | 1.3616   | 0.2189 | 0.8254     | -1204  | 860      |
| 80  | 0.9872 | 0.5295     | 3.5493     | 5.3044 | 0.1839   | 1.3596   | 0.2190 | 0.8400     | -1977  | 1465     |
| 100 | 0.9879 | 0.5293     | 3.5537     | 5.3029 | 0.1846   | 1.3577   | 0.2189 | 0.8434     | -2397  | 2289     |
| 200 | 0.9898 | 0.5285     | 3.5647     | 5.2997 | 0.1854   | 1.3493   | 0.2186 | 0.8480     | -4715  | 5873     |

**Table E.8.** Multi-Step ML Estimation.  $\theta_0 = 0.9 \cdot \theta$ . Model (E.68).

| $j$ | $K$    | $\alpha_1$ | $\alpha_2$ | $\tau$ | $\sigma$ | $\gamma$ | $r$    | $\sigma/r$ | $V$    | CPU sec. |
|-----|--------|------------|------------|--------|----------|----------|--------|------------|--------|----------|
| 1   | 0.9797 | 0.5651     | 3.4211     | 5.3164 | 0.2204   | 1.3762   | 0.2077 | 1.0613     | -63.27 | 111      |
| 4   | 0.9798 | 0.5607     | 3.4251     | 5.3186 | 0.2182   | 1.4745   | 0.2102 | 1.0383     | -248.7 | 160      |
| 8   | 0.9798 | 0.5307     | 3.4369     | 5.3425 | 0.2051   | 1.5453   | 0.2140 | 0.9581     | -468.5 | 237      |
| 20  | 0.9801 | 0.5262     | 3.4700     | 5.3392 | 0.1860   | 1.5200   | 0.2189 | 0.8498     | -829.0 | 490      |
| 40  | 0.9835 | 0.5302     | 3.5242     | 5.3124 | 0.1807   | 1.3614   | 0.2189 | 0.8254     | -1204  | 878      |
| 80  | 0.9872 | 0.5294     | 3.5495     | 5.3043 | 0.1840   | 1.3603   | 0.2190 | 0.8403     | -1977  | 1616     |
| 100 | 0.9879 | 0.5295     | 3.5535     | 5.3027 | 0.1843   | 1.3535   | 0.2189 | 0.8421     | -2397  | 2508     |
| 200 | 0.9898 | 0.5298     | 3.5640     | 5.2987 | 0.1840   | 1.3310   | 0.2185 | 0.8420     | -4715  | 6730     |

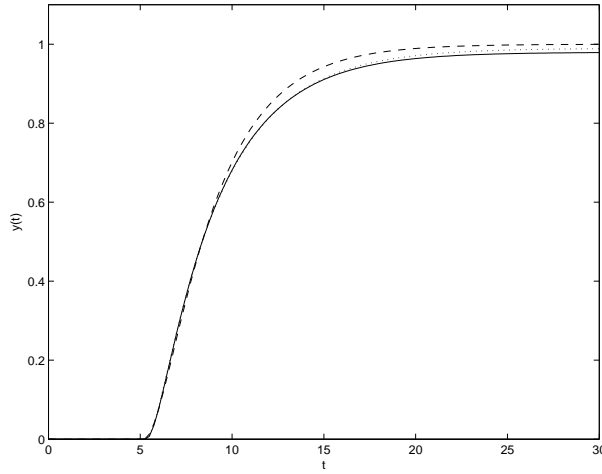


**Figure E.1.** IO-data for the SISO system,  $\mathcal{S}$ , defined by (E.66)-(E.67). The inputs,  $\{u(t)\}$ , are PRBS with bandwidth  $[0 \ 0.02]$  and levels  $[-1 \ 1]$ .

The estimates for the single-step and multi-step least squares criteria, various prediction horizons and various starting guesses of the parameters are shown in tables E.1-E.4.<sup>1</sup> From these results, it is apparent that the LS method cannot be used to uniquely estimate  $\sigma$  and  $r$ . However, their ratio seems to be constant for different starting guesses and decreases with increasing horizon. This implies that the identified model approaches an output error model for long prediction horizons.

The estimates for the single-step and multi-step maximum likelihood criteria, various prediction horizons and various starting guesses of the parameters are shown in tables E.5-E.8.  $\sigma$  and  $r$  are estimated consistently for various initial guesses. For long-range single-step maximum likelihood estimation, the estimated model is essentially an output error model. The step response for the true model and the models estimated by the multi-step maximum-likelihood criterion with prediction horizons  $N_p = 1$  and  $N_p = 200$  are shown in figure E.2. There is not much difference between the two estimated models, but a little steady difference compared to the true model. However, as can be read off from tables E.7-E.8 the main difference between the estimated models for prediction horizon  $N_p = 1$  and prediction horizon  $N_p = 200$  is not the deterministic transfer function,  $\hat{g}(s)$ , but the disturbance model,  $\hat{h}(s)$ , and the covariance of the measurement noise,  $\hat{r}^2$ .

<sup>1</sup>All computations are conducted using a 3.20 GHz Pentium IV processor. The CPU time is reported to indicate the order of magnitude of computing time needed to calculate the various estimates.



**Figure E.2.** Step response for the deterministic part of the SISO model estimated using a simplified model with an output integrator. Estimated model (E.68a) using the multi-step maximum likelihood criterion with  $N_p = 1$  (solid line) and  $N_p = 200$  (dotted line). Dashed line: True model (E.67a).

### E.5.2 Simplified Model with Output Integrator

In this subsection we will illustrate the methodology when the model structure,  $\mathcal{M}$ , is different from the system model,  $\mathcal{S}$ , used to generate the data, i.e.  $\mathcal{S} \notin \mathcal{M}$ . To do this consider the model

$$\hat{g}(s) = \frac{\hat{K}}{\hat{\alpha}s + 1} e^{-\hat{\tau}s} \quad (\text{E.69a})$$

$$\hat{h}(s) = \frac{\hat{\sigma}}{s} \quad (\text{E.69b})$$

In the process industries most stable models can be approximated quite well by delayed first-order transfer functions,  $\hat{g}(s)$ . The disturbance model,  $\hat{h}(s)$ , is chosen as an integrator to ensure off-set free control for step-type disturbances and model-plant mismatch in the resulting predictive control system for which the estimated model is applied. For internal model control (IMC) which can be considered as a restricted class of predictive control this modelling approach is commonplace (Morari and Zafiriou, 1989).

In contrast to the least-squares prediction-error-methods, the maximum-likelihood prediction-error-methods yield unique estimates for the covariance matrices. Hence, only the maximum-likelihood prediction-error-estimates for the system (E.69) will be reported here. The single-step maximum-likelihood estimates for various prediction horizons are shown in table E.9. As the prediction horizon increases,  $\hat{\sigma}$  is decreased and the estimated model becomes essentially an out-

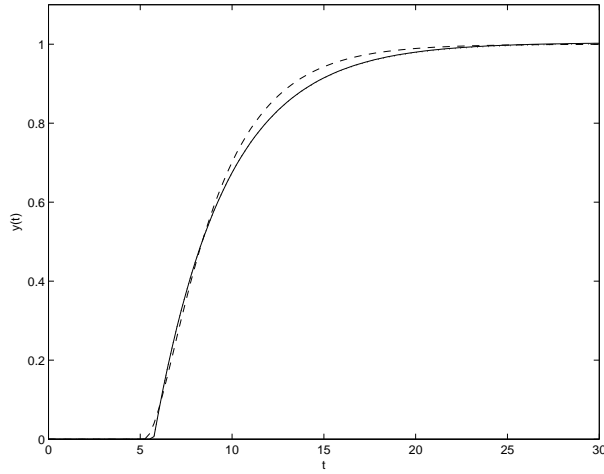
**Table E.9.** Single-Step ML Estimation. Model (E.69).

| $j$ | $K$    | $\alpha$ | $\tau$ | $\sigma$ | $r$    | $\sigma/r$ | $V$    | CPU sec. |
|-----|--------|----------|--------|----------|--------|------------|--------|----------|
| 1   | 1.0043 | 3.8386   | 5.7243 | 0.0658   | 0.2226 | 0.2959     | -19.78 | 123      |
| 4   | 0.9911 | 3.6390   | 5.7547 | 0.0124   | 0.2424 | 0.0511     | 30.24  | 399      |
| 8   | 0.9811 | 3.5585   | 5.7792 | 0.0006   | 0.2490 | 0.0025     | 34.02  | 857      |
| 20  | 0.9812 | 3.5568   | 5.7802 | 0.0004   | 0.2455 | 0.0018     | 29.20  | 2245     |
| 40  | 0.9822 | 3.5750   | 5.7697 | 0.0002   | 0.2479 | 0.0009     | 48.26  | 4192     |
| 80  | 0.9747 | 3.5664   | 5.7618 | 0.0002   | 0.2547 | 0.0008     | 102.8  | 7448     |
| 100 | 1.0107 | 3.6458   | 5.6487 | 0.0002   | 0.3169 | 0.0006     | 539.5  | 8212     |
| 200 | 0.9465 | 3.3713   | 5.8331 | 0.0006   | 0.2556 | 0.0023     | 110.5  | 17885    |

**Table E.10.** Multi-Step ML Estimation. Model (E.69).

| $j$ | $K$    | $\alpha$ | $\tau$ | $\sigma$ | $r$    | $\sigma/r$ | $V$    | CPU sec. |
|-----|--------|----------|--------|----------|--------|------------|--------|----------|
| 1   | 1.0043 | 3.8386   | 5.7243 | 0.0658   | 0.2226 | 0.2959     | -19.78 | 120      |
| 4   | 1.0043 | 3.8387   | 5.7244 | 0.0659   | 0.2424 | 0.2962     | -79.72 | 160      |
| 8   | 1.0043 | 3.8386   | 5.7243 | 0.0658   | 0.2490 | 0.2956     | -161.4 | 257      |
| 20  | 1.0043 | 3.8389   | 5.7244 | 0.0660   | 0.2455 | 0.2968     | -398.4 | 382      |
| 40  | 1.0044 | 3.8394   | 5.7245 | 0.0666   | 0.2479 | 0.2995     | -780.4 | 722      |
| 80  | 1.0039 | 3.8319   | 5.7256 | 0.0669   | 0.2547 | 0.3011     | -1541  | 1550     |
| 100 | 1.0033 | 3.8277   | 5.7261 | 0.0670   | 0.3169 | 0.3018     | -1954  | 2082     |
| 200 | 1.0024 | 3.8209   | 5.7269 | 0.0672   | 0.2556 | 0.3027     | -4234  | 4268     |

put error model. For the case considered, the estimated process noise vanishes already at a prediction horizon of  $j = 8$ . The measurement noise is increased slightly as the prediction horizon increases to accommodate the output noise that is not caught by the process noise. The multi-step maximum-likelihood estimates are shown in table E.10. Compared to the single-step maximum likelihood estimates the multi-step maximum-likelihood estimates are much less sensitive to the chosen prediction horizon. In fact there is not much difference between the estimated parameters for the one-step ahead maximum likelihood estimate and the multi-step maximum likelihood estimate with a very long prediction horizon, i.e.  $N_p = 200$ . The step responses for the estimated multi-step maximum likelihood estimate with a prediction horizon of  $N_p = 1$ , i.e. the one-step maximum likelihood estimate, and a prediction horizon of  $N_p = 200$  are shown in figure E.3. They can hardly be distinguished. Hence, for all practical purposes they can be considered identical. This suggests that the one-step ahead prediction maximum-likelihood estimate should be applied in practice as the computing time for the one-step ahead prediction maximum-likelihood estimate is considerably lower than the computing time for the multi-step maximum-likelihood prediction with a long prediction horizon ( $N_p = 200$ ). Figure E.3 also depicts the step response of the true system.



**Figure E.3.** Step response for the deterministic part of the SISO model estimated using a simplified model with an output integrator. Solid line: Estimated model (E.69a) using the multi-step maximum likelihood criterion with  $N_p = 1$  and  $N_p = 200$ . Dashed line: True model (E.67a).

It is evident that the step responses of the estimated models approximate the true step response quite well.

## E.6 Wood and Berry Distillation Example

Wood and Berry (Wood and Berry, 1973; Ogunnaike and Ray, 1994) propose the following model for a distillation column separating methanol and water

$$Y(s) = G(s)U(s) + G_d(s)D(s) \quad (\text{E.70})$$

with

$$G(s) = \begin{bmatrix} \frac{12.8e^{-s}}{16.7s+1} & \frac{-18.9e^{-3s}}{21.0s+1} \\ \frac{6.6e^{-7s}}{10.9s+1} & \frac{-19.4e^{-3s}}{14.4s+1} \end{bmatrix} \quad (\text{E.71a})$$

$$G_d(s) = \begin{bmatrix} \frac{3.8e^{-8.1s}}{14.9s+1} \\ \frac{4.9e^{-3.4s}}{13.2s+1} \end{bmatrix} \quad (\text{E.71b})$$

The variables in the model are:  $y_1$  is the overhead methanol mole fraction,  $y_2$  is the bottom product methanol mole fraction,  $u_1$  is the overhead reflux flow rate,  $u_2$  is the bottoms steam flow rate, and  $d$  is the column feed flow rate.

The output data for the Wood and Berry distillation column is generated using

the stochastic model

$$\mathbf{Z}(s) = G(s)U(s) + G_d(s)(D(s) + \sigma \mathbf{E}(s)) \quad (\text{E.72a})$$

$$\mathbf{y}(t_k) = \mathbf{z}(t_k) + \mathbf{v}(t_k) \quad (\text{E.72b})$$

with  $\mathbf{E}(s)$  being white noise and  $\sigma = 1$ . The measurement noise is

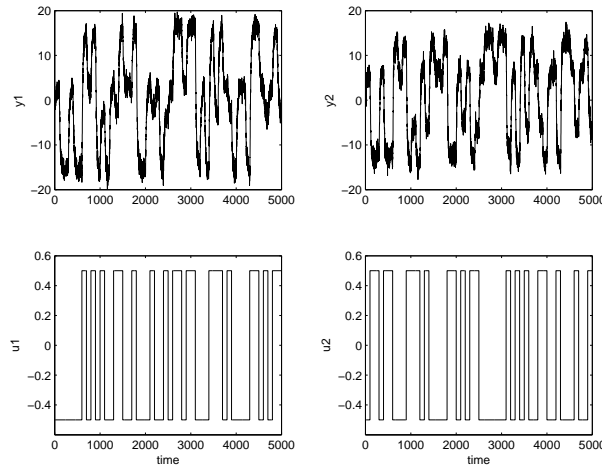
$$\mathbf{v}(t_k) \sim N_{iid} \left( \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} r_1^2 & 0 \\ 0 & r_2^2 \end{bmatrix} \right) \quad (\text{E.73})$$

in which  $r_1 = r_2 = 1.0$ . The sampling time of the system is  $T_s = 1.0$ . In the identification experiment  $u$  is a pseudo random binary sequence and the systematic feed flow rate deviation,  $d$ , is set to zero.

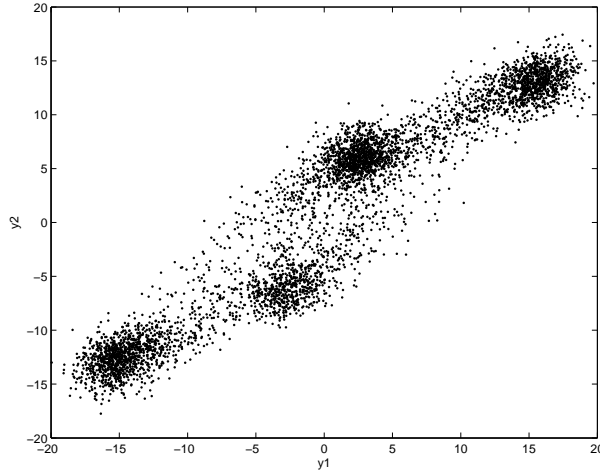
The IO-data generated for this system and used for identification is plotted in figure E.4. It is apparent that outputs are highly co-linear, i.e. the system is ill-conditioned. This is a well known phenomenon for distillation columns and may in many cases require closed-loop identification or specially designed perturbations suitable for the directionality of the plant (Andersen *et al.*, 1989, 1991; Andersen and Kümmel, 1992a,b; Koung and MacGregor, 1994; Zhu, 2001).

### E.6.1 Identification of a Control Relevant Model

In the following the generated open-loop data will be used for estimation of a process model,  $\hat{G}(s)$ , and a disturbance model,  $\hat{H}(s)$ . The application of this model for predictive control is demonstrated.



**Figure E.4.** IO-data for the Wood and Berry distillation column simulated using the model defined by (E.72) with the transfer functions (E.71).  $\sigma = 1$  and  $r_1 = r_2 = 1.0$ . The inputs,  $\{u(t)\}$ , are PRBS with bandwidth  $[0 \ 0.01]$  and levels  $[-0.5 \ 0.5]$ .



**Figure E.5.** Phase plane plot of the output data of the Wood and Berry distillation column simulated using the model defined by (E.72) with the transfer functions (E.71).  $\sigma = 1$  and  $r_1 = r_2 = 1.0$ . The inputs,  $\{u(t)\}$ , are PRBS with bandwidth  $[0 \ 0.01]$  and levels  $[-0.5 \ 0.5]$ . The figure shows that the output data are highly co-linear.

The estimated model is of the form

$$\hat{\mathbf{Z}}(s) = \hat{G}(s)U(s) + \hat{H}(s)\hat{\mathbf{E}}(s) \quad (\text{E.74a})$$

$$\mathbf{y}(t_k) = \hat{\mathbf{z}}(t_k) + \hat{\mathbf{v}}(t_k) \quad (\text{E.74b})$$

in which  $\hat{G}(s)$  is a transfer function with the same structure as  $G(s)$ . The disturbance model,  $\hat{H}(s)$ , has the structure

$$H(s) = \begin{bmatrix} h_{11}(s) & 0 \\ 0 & h_{22}(s) \end{bmatrix} \quad (\text{E.75})$$

with

$$h_{ii}(s) = \frac{1}{s} \frac{\sigma_{ii}}{\gamma_{ii}s + 1} \quad i = 1, 2 \quad (\text{E.76})$$

Note that the disturbance model is equipped with integrators in order to ensure steady-state offset-free control when the model is applied in a predictive controller (Muske and Badgwell, 2002; Pannocchia and Rawlings, 2003). Using a multi-step (one-step) maximum likelihood criterion with prediction horizon



$N_p = 1$ , the estimated model transfer functions are

$$\hat{G}(s) = \begin{bmatrix} \frac{13.21e^{-0.84s}}{17.20s+1} & \frac{-18.52e^{-3.34s}}{20.67s+1} \\ \frac{6.72e^{-7.69s}}{10.03s+1} & \frac{-19.28e^{-3.07s}}{14.77s+1} \end{bmatrix} \quad (\text{E.77a})$$

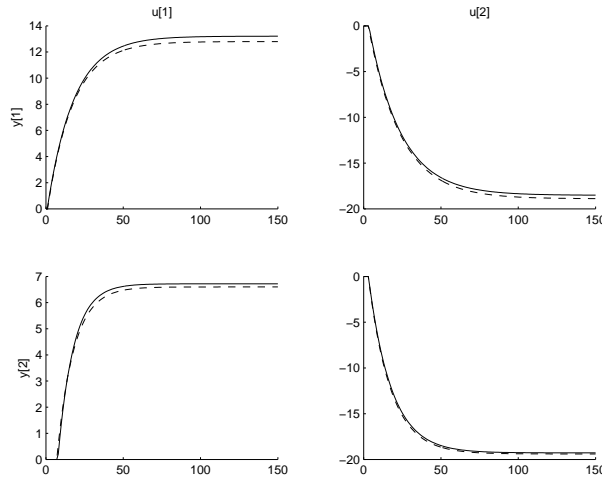
$$\hat{H}(s) = \begin{bmatrix} \frac{1}{s} \frac{0.18}{0.16s+1} & 0 \\ 0 & \frac{1}{s} \frac{0.27}{0.16s+1} \end{bmatrix} \quad (\text{E.77b})$$

and the estimated covariance of the measurement noise is

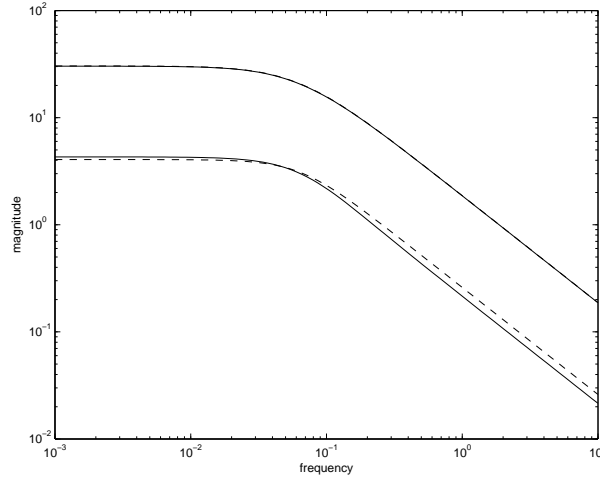
$$\hat{R}_{vv} = \begin{bmatrix} 1.03^2 & 0 \\ 0 & 1.04^2 \end{bmatrix} \quad (\text{E.77c})$$

In this particular disturbance model, we have not utilized that the impact of the actual disturbance on the outputs are correlated even though this is evident from the estimated disturbance model. This implies that the estimated model is essentially two MISO models.

The step responses of the estimated transfer function,  $\hat{G}(s)$ , and the true transfer function,  $G(s)$ , are shown in figure E.6. It is evident that the estimated model represents the true model well. To illustrate the identification consequences of having co-linear output data as shown in figure E.5, the high gain and low gain direction gains are plotted as function of frequency in figure E.7. It is evident that the gain in the high gain direction is estimated accurately, while there is some error in the estimate of the gain in the low gain direction. This phenomena is well known for ill-conditioned plants and can be overcome by closed-loop identification or identification experiments taking the high-gain and low-gain direction into account.



**Figure E.6.** Step responses for the deterministic part of the Wood and Berry distillation column model. Solid line: Estimated model. Dashed line: True model.

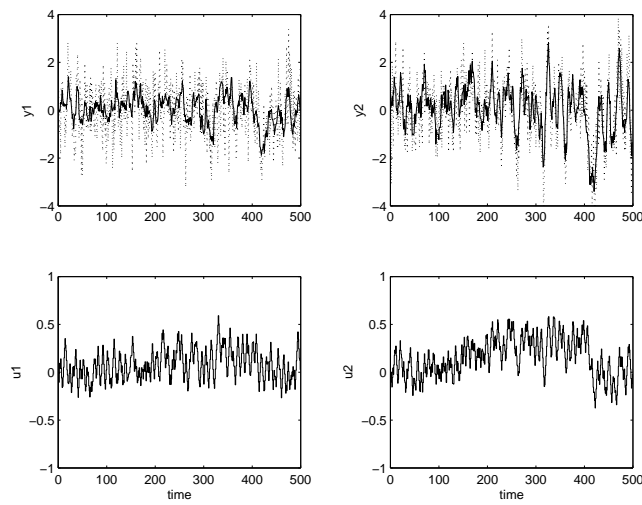


**Figure E.7.** Low and high gain plots for the true model (dashed line) and the estimated model (solid line) of the Wood and Berry distillation column. It is evident that the error of the estimated model is almost entirely in the low gain direction, while the gain in the high gain direction is estimated accurately.

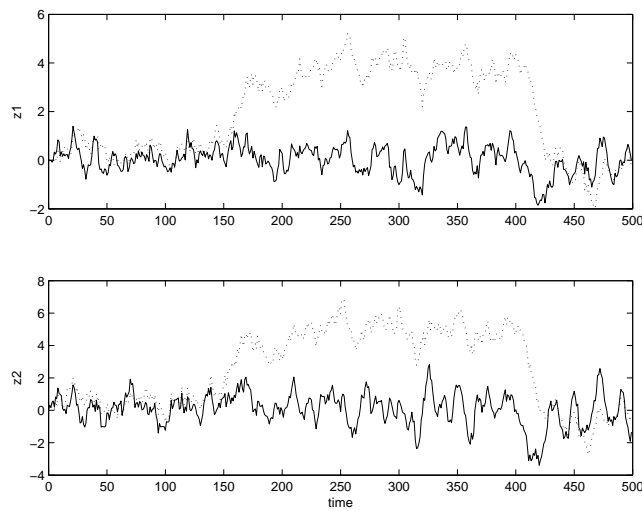
### E.6.2 Model Predictive Control

The suitability of the proposed identification method for predictive control is validated by application of the identified model (E.77) in the design of a constrained multivariable predictive controller. This controller is tested in a simulation using (E.72) as the plant. The transfer functions are defined by (E.71),  $\sigma = 1$  and the measurement noise covariance is defined by (E.73). At time  $t = 150$  a deterministic feed flow step disturbance,  $d = 1$ , occurs. This disturbance is unknown to the controller. At time  $t = 400$ , the disturbance disappears again.

The performance of the model predictive controller is shown in figure E.8. In the upper plots the noise free outputs,  $z_1$  and  $z_2$ , as well as the measurements,  $y_1$  and  $y_2$ , are shown. The measurement noise is significant. Its effect on the measured output is of the same order of magnitude as the effect of the process noise. This corresponds to the common industrial plant using low resolution sensors contaminated with a high level of measurement noise. In this situation, no control system can completely eliminate the effect of the process noise on the outputs. However, as is evident by figure E.9 which compares the open-loop outputs, ( $z_1$  and  $z_2$ , dotted line), to the closed-loop outputs, ( $z_1$  and  $z_2$ , solid line), the controller rejects the disturbance and performs marginally better than the no-control (open-loop) case in the situation with only white process noise. This closed-loop performance of the constrained predictive con-



**Figure E.8.** The Wood and Berry distillation column controlled by a predictive controller. Top: Outputs ( $z$ , solid line) and measured outputs ( $y$ , dotted line). Bottom: Controlled process inputs,  $u$ .



**Figure E.9.** The process outputs,  $z$ , for the Wood and Berry distillation column controlled by a predictive controller (solid line) and the identical scenario without control (dotted line).

troller indicates that the proposed prediction-error- methodology is suitable and feasible for identification of models that can be used by a predictive controller. For the performance of the controller in the situation with model-plant mismatch as well as unmeasured disturbance, it is important that the noise model is equipped with integrators. In addition identification of parameters in stochastic transfer functions with delays have proven feasible and very useful for specification of predictive controllers.

## E.7 Conclusion

A constructive method for estimation of parameters in continuous-discrete-time stochastic systems described by transfer functions with time delays has been described and demonstrated. The method applies prediction-error criteria and the predictions are generated using the Kalman filter and predictor for a stochastic linear discrete-time state space model equivalent to the continuous-discrete-time stochastic transfer function model with time delays. In particular, an efficient computing scheme for the multi-step maximum likelihood prediction-error estimator is developed. The multi-step prediction-error criteria may be selected such that they are compatible with the optimization criterion applied by the predictive controller that uses the identified model. Compared to the single-step least-squares and the single-step maximum likelihood estimators, the multi-step maximum likelihood estimator produces parameter estimates that are less sensitive to the prediction horizon applied. In contrast to the single-step and multi-step least squares estimators, the multi-step maximum likelihood estimator computes unique parameters for the process and measurement noise. Hence, the multi-step maximum likelihood estimators are recommended for predictive control. Depending on the prediction horizon, the multi-step maximum likelihood estimator requires much more computer resources than the single-step one-step ahead least-squares predictor.

A number of authors (Shook *et al.*, 1992; Gopaluni *et al.*, 2003, 2004; Ljung, 1999) recommend that the pre-filter for the prediction errors is chosen in accordance with the prediction horizon in the predictive controller. However, our SISO simulation example suggests that the multi-step maximum likelihood estimates depend only weakly on the prediction horizon. The reason for this apparent contradiction between our simulation example and conventional wisdom is that we apply a rigorous maximum-likelihood estimator, while traditional prediction-error methods (Shook *et al.*, 1992; Gopaluni *et al.*, 2003, 2004; Ljung, 1999) neglect the covariance of the multi-step prediction errors.

Consequently, based on the SISO simulation example, *we recommend the maximum likelihood (or maximum a posteriori) estimator based on the one-step-ahead prediction-error*. The models obtained using the multi-step maximum-likelihood prediction-error method with a prediction horizon of one and a very long prediction horizon are essentially identical. However, the long prediction

horizon demands much more computational resources than the criterion based on the one-step-ahead prediction.

The feasibility of the suggested approach for predictive control is demonstrated using the Wood and Berry (Wood and Berry, 1973) distillation example. In particular, the design of predictive controllers using continuous-discrete-time stochastic models specified by transfer functions with delays has proven very convenient.

## E.8 Array Algorithms

The array algorithm implementations of the Kalman filter propagates the matrix square root (a lower triangular matrix) of the covariance matrices. Thereby, they circumvent the problems in the ordinary Kalman filter recursions about maintaining symmetry and positive definiteness. These problems may arise due to finite precision and round-off for ill-conditioned systems (Kaminski *et al.*, 1971; Morf and Kailath, 1975; Bierman, 1977; Bierman and Thornton, 1977; Verhaegen and Van Dooren, 1986; Kailath *et al.*, 2000). Problems with ill-conditioning are particularly likely to occur when the Kalman estimators are applied within an optimization algorithm for parameter estimation. The reason for this is that the parameters may vary dramatically during the course of an optimization.

The array algorithms proceed by transforming a pre-array,  $\mathcal{A}$ , into a lower triangular post-array,  $[L \ 0]$ , using orthogonal transformations,  $\Theta$ , i.e.

$$\mathcal{A}\Theta = [L \ 0] \quad (\text{E.78})$$

The orthogonal transformations may be conducted using *Householder transformations*, *Givens rotations* or *fast Givens transformations* (Golub and Van Loan, 1996; Kailath *et al.*, 2000). For the general treatment of this problem Householder transformation are most efficient, while Givens rotations and fast Givens transformation are more efficient when the problem is highly structured.

The ordinary Kalman-filter implementation may be considered as the dynamic state space equivalent of the normal equations in least squares problems. Similarly, the array methods may be regarded as the dynamic state space equivalent of the QR-method for solution of least squares problems. In this appendix, the pre-arrays and the structure of the post-arrays are presented and their relevance to the prediction error methods is indicated.

### E.8.1 One-Step Predictions

The one-step prediction error and the associated covariances for the model (E.1) are presented and it is outlined how the results are used in the prediction error algorithms. The resulting algorithm corresponds to (E.2)-(E.6a) in the

ordinary Kalman filter implementation. The pre- and post-array can be stated as

$$\begin{bmatrix} R_{vv}^{1/2} & CP_{k|k-1}^{1/2} & 0 \\ R_{wv}R_{vv}^{-T/2} & AP_{k|k-1}^{1/2} & Q_s^{1/2} \end{bmatrix} \Theta = \begin{bmatrix} X & 0 & 0 \\ Y & Z & 0 \end{bmatrix} \quad (\text{E.79})$$

in which  $Q_s = R_{ww} - R_{wv}R_{vv}^{-1}R'_{wv}$  and  $\Theta$  is some orthogonal transformation. Note that the algorithm simplifies considerably when  $R_{wv} = 0$ . The post-array matrices are

$$X = R_{e,k}^{1/2} \quad (\text{E.80a})$$

$$Y = (AP_{k|k-1}C' + R_{wv})R_{e,k}^{-T/2} \quad (\text{E.80b})$$

$$Z = P_{k+1|k}^{1/2} \quad (\text{E.80c})$$

Hence,  $Z = P_{k+1|k}^{1/2}$  is the updated one-step state prediction-error square root covariance.  $X$  and  $Y$  may be used in computing the one step prediction error,  $\varepsilon_{k|k-1} = e_k$ , and to update the one-step state prediction,  $\hat{x}_{k+1|k}$ , i.e.

$$\hat{y}_{k|k-1} = C\hat{x}_{k|k-1} \quad (\text{E.81a})$$

$$e_k = y_k - \hat{y}_{k|k-1} \quad (\text{E.81b})$$

$$\bar{e}_k = X^{-1}e_k \quad (\text{E.81c})$$

$$\hat{x}_{k+1|k} = A\hat{x}_{k|k-1} + Bu_k + Y\bar{e}_k \quad (\text{E.81d})$$

Note that  $X$  is a lower triangular matrix. Hence,  $\bar{e}_k = X^{-1}e_k$  may be computed without factorizing  $X$  as it is already factorized. The one-step least squares prediction error objective function apply  $\varepsilon_{k|k-1} = e_k$  and the contribution to the objective function may be computed as  $\varepsilon'_{k|k-1}\varepsilon_{k|k-1} = e'_ke_k$ . The contribution at time  $k$  to the maximum likelihood objective function may be computed efficiently as

$$\varepsilon'_{k|k-1}R_{e,k}^{-1}\varepsilon_{k|k-1} = \bar{e}'_k\bar{e}_k \quad (\text{E.82})$$

Since  $X = R_{e,k}^{1/2}$  is lower triangular, the determinant,  $\det R_{k|k-1}$ , may be computed as

$$\det R_{k|k-1} = \det R_{e,k} = \prod_{i=1}^{n_y} X_{i,i}^2 = \left( \prod_{i=1}^{n_y} X_{i,i} \right)^2 \quad (\text{E.83})$$

## E.8.2 j-Step Predictions

For  $j$ -step prediction error maximum likelihood methods, the covariances,  $P_{k+j|k} = P_{k+j|k}^{1/2}P_{k+j|k}^{T/2}$  may be propagated using the following array algorithm for the square root

$$\begin{bmatrix} AP_{k+j|k}^{1/2} & R_{ww}^{1/2} \end{bmatrix} \Theta = \begin{bmatrix} P_{k+j+1|k}^{1/2} & 0 \end{bmatrix} \quad (\text{E.84})$$

in which  $\Theta$  is an orthogonal transformation. Similarly, the square roots in  $R_{k+j|k} = R_{k+j|k}^{1/2} R_{k+j|k}^{T/2}$  may be computed using an array algorithm

$$\begin{bmatrix} CP_{k+j|k}^{1/2} & R_{vv}^{1/2} \end{bmatrix} \Theta = \begin{bmatrix} R_{k+j|k}^{1/2} & 0 \end{bmatrix} \quad (\text{E.85})$$

in which  $\Theta$  is an orthogonal transformation. For the maximum likelihood  $j$ -step prediction error method, the needed determinant may be computed as

$$\det R_{k+j|k} = \prod_{i=1}^{n_y} (R_{k+j|k}^{1/2})_{i,i}^2 = \left( \prod_{i=1}^{n_y} (R_{k+j|k}^{1/2})_{i,i} \right)^2 \quad (\text{E.86})$$

# References

- AEA Technology (1995). *Harwell Subroutine Library. Release 12*. AEA Technology, Harwell Laboratory, Oxfordshire, England.
- Allgöwer, F. and Zheng, A., editors (2000). *Nonlinear Model Predictive Control*. Birkhäuser, Basel.
- Allgöwer, F.; Badgwell, T. A.; Qin, J. S.; Rawlings, J. B. and Wright, S. J. (1999). Nonlinear Predictive Control and Moving Horizon Estimation - An Introductory Overview. In P. M. Frank, editor, *Advances in Control. Highlights of ECC'99*, pages 391–449. Springer.
- Andersen, E. D. (2000). On Primal and Dual Infeasibility Certificates in a Homogeneous Model for Convex Optimization. *SIAM Journal of Optimization*, **11**, 380–388.
- Andersen, E. D. and Andersen, K. D. (2000). The Mosek Interior Point Optimizer for Linear Programming: An Implementation of the Homogenous Algorithm. In H. Frenk; K. Roos; T. Terlaky and S. Zhang, editors, *High-Performance Optimization*, pages 197–232. Kluwer Academic Publishers, Boston.
- Andersen, E. D. and Ye, Y. (1998). A Computational Study of the Homogeneous Algorithm for Large-Scale Convex Optimization. *Computational Optimization and Applications*, **10**, 243–269.
- Andersen, E. D. and Ye, Y. (1999). On a Homogeneous Algorithm for the Monotone Complementarity Problem. *Mathematical Programming*, **84**, 375–399.
- Andersen, E. D.; Roos, C. and Terlaky, T. (2003). On Implementing a Primal-Dual Interior Point Method for Conic Quadratic Optimization. *Mathematical Programming*, **95**.
- Andersen, H. W. (1989). *Advanced Control of Distillation Columns*. Ph.D. thesis, Technical University of Denmark.
- Andersen, H. W. and Kümmel, M. (1992a). Evaluating Estimation of Gain Directionality. Part 1: Methodology. *Journal of Process Control*, **2**, 59–66.
- Andersen, H. W. and Kümmel, M. (1992b). Evaluating Estimation of Gain Directionality - Part 2: A Case Study of Binary Distillation. *Journal of Process Control*, **2**, 67–86.



- Andersen, H. W.; Kümmel, M. and Jørgensen, S. B. (1989). Dynamics and Identification of a Binary Distillation Column. *Chemical Engineering Science*, **44**, 2571–2581.
- Andersen, H. W.; Rasmussen, K. H. and Jørgensen, S. B. (1991). Advances in Process Identification. In W. H. Ray and Y. Arkun, editors, *Chemical Process Control IV*, pages 237–269, New York, USA. AIChE.
- Andersen, T. R. (2002). *Optimal Design and Operation of Process Integrated Distillation*. Ph.D. thesis, Department of Chemical Engineering, Technical University of Denmark.
- Anderson, E.; Bai, Z.; Bischof, C.; Blackford, S.; Demmel, J.; Dongarra, J.; Croz, J. D.; Greenbaum, A.; Hammarling, S.; McKenney, A. and Sorensen, D. (1999). *LAPACK User's Guide*. SIAM, Philadelphia, PA, third edition.
- Arrow, K. J. (1951). Alternative Approaches to the Theory of Choice in Risk-Taking Situations. *Econometrica*, **19**, 404–437.
- Arrow, K. J.; Blackwell, D. and Girshick, M. A. (1949). Bayes and Minimax Solutions of Sequential Design Problems. *Econometrica*, **17**, 213–244.
- Artstein, Z. and Leizarowitz, A. (1985). Tracking Periodic Signals with the Overtaking Criterion. *IEEE Transactions on Automatic Control*, **AC-30**, 1123–1126.
- Åström, K. J. (1970a). *Introduction to Stochastic Control Theory*. Academic Press.
- Åström, K. J. (1970b). *Introduction to Stochastic Control Theory*. Academic Press, New York.
- Åström, K. J. (1980). Maximum Likelihood and Prediction Error Methods. *Automatica*, **16**, 551–574.
- Åström, K. J. and Wittenmark, B. (1997). *Computer Controlled Systems, Theory and Design*. Prentice-Hall, Englewood Cliffs, N.J.
- Bar-Shalom, Y. (1981). Stochastic Dynamic Programming: Caution and Prob-ing. *IEEE Transactions on Automatic Control*, **26**, 1184–1195.
- Barbieri, E. and Alba-Flores, R. (2000). On the Infinite-Horizon LQ Tracker. *Systems & Control Letters*, **40**, 77–82.
- Bard, Y. (1974). *Nonlinear Parameter Estimation*. Academic Press, New York.
- Bartlett, R. A.; Wächter, A. and Biegler, L. T. (2000). Active Set Vs. Interior Point Strategies for Model Predictive Control. In *American Control Conference 2000*, pages 4229–4233, Chicago, Illinois.

- Bartlett, R. A.; Biegler, L. T.; Backstrom, J. and Gopal, V. (2002). Quadratic Programming Algorithms for Large-Scale Model Predictive Control. *Journal of Process Control*, **12**, 775–795.
- Bauer, I. (2000). *Numerische Verfahren Zur Lösung Von Anfangswertaufgaben und Zur Generierung Von Ersten und Zweiten Ableitungen mit Anwendungen Bei Optimierungsaufgaben in Chemie und Verfahrenstechnik*. Ph.D. thesis, University of Heidelberg.
- Bellman, R. E. and Dreyfus, S. E. (1962). *Applied Dynamic Programming*. Princeton University Press.
- Bemporad, A.; Morari, M.; Dua, V. and Pistikopoulos, E. N. (2002). The Explicit Linear Quadratic Regulator for Constrained Systems. *Automatica*, **38**, 3–20.
- Bertsekas, D. P. (1972). Infinite-Time Reachability of State-Space Regions by Using Feedback Control. *IEEE Trans. Automat. Contr.*, **17**, 604–613.
- Bertsekas, D. P. (1976). *Dynamic Programming and Stochastic Control*. Academic Press, New York.
- Bertsekas, D. P. (1995a). *Dynamic Programming and Optimal Control*, volume 1. Athena Scientific, Belmont, Massachusetts.
- Bertsekas, D. P. (1995b). *Nonlinear Programming*. Athena Scientific, Belmont, Massachusetts.
- Bertsekas, D. P. and Rhodes, I. B. (1971a). On the Minmax Reachability of Target Set and Target Tubes. *Automatica*, **7**, 233–247.
- Bertsekas, D. P. and Rhodes, I. B. (1971b). Recursive State Estimation for a Set-Membership Description of Uncertainty. *IEEE Trans. Automat. Contr.*, **16**, 117–128.
- Betts, J. T. (2001). *Practical Methods for Optimal Control Using Nonlinear Programming*. SIAM, Philadelphia.
- Betts, J. T. and Huffman, W. P. (1998). Mesh Refinement in Direct Transcription Methods for Optimal Control. *Optimal Control Applications and Methodologies*, **19**, 1–21.
- Biegler, L. T. (1984). Solution of Dynamic Optimization Problems by Successive Quadratic Programming and Orthogonal Collocation. *Computers and Chemical Engineering*, **8**, 243–248.
- Biegler, L. T. (1998). Advances in Nonlinear Programming Concepts for Process Control. *Journal of Process Control*, **8**, 301–311.

- Biegler, L. T. (2000). Efficient Solution of Dynamic Optimization and NMPC Problems. In F. Allgöwer and A. Zheng, editors, *Nonlinear Model Predictive Control*, pages 219–243. Birkhäuser, Basel, Switzerland.
- Biegler, L. T. and Cuthrell, J. E. (1985). Improved Infeasible Path Optimization for Sequential Modular Simulators - II: The Optimization Algorithm. *Computers and Chemical Engineering*, **8**, 257–267.
- Bierman, G. J. (1977). *Factorization Methods for Discrete Sequential Estimation*. Academic Press, NY.
- Bierman, G. J. and Thornton, C. L. (1977). Numerical Comparison of Kalman Filter Algorithms: Orbit Determination Case Study. *Automatica*, **13**, 23–35.
- Binder, T. (2002). *Adaptive Multiscale Methods for the Solution of Dynamic Optimization Problems*. Ph.D. thesis, Lehrstuhl für Prozesstechnik, RWTH Aachen.
- Binder, T.; Blank, L.; Dahmen, W. and Marquardt, W. (2000). An Adaptive Multiscale Method for Real-Time Moving Horizon Optimization. In *Proceedings of the American Control Conference*, pages 4234–4238, Chicago, Illinois.
- Binder, T.; Blank, L.; Bock, H. G.; Burlisch, R.; Dahmen, W.; Diehl, M.; Kronseder, T.; Marquardt, W.; Schlöder, J. P. and von Stryk, O. (2001a). Introduction to Model Based Optimization of Chemical Processes on Moving Horizons. In M. Grötschel; S. O. Krumke and J. Rambau, editors, *Online Optimization of Large Scale Systems*, pages 295–339. Springer, Berlin.
- Binder, T.; Blank, L.; Dahmen, W. and Marquardt, W. (2001b). Iterative Algorithms for Multiscale State Estimation, Part 1: Concepts. *Journal of Optimization Theory and Applications*, **111**, 501–527.
- Binder, T.; Blank, L.; Dahmen, W. and Marquardt, W. (2001c). Iterative Algorithms for Multiscale State Estimation, Part 2: Numerical Investigations. *Journal of Optimization Theory and Applications*, **111**, 529–551.
- Birge, J. R. and Louveaux, F. (1997). *Introduction to Stochastic Programming*. Springer, New York.
- Bitmead, R. R.; Gevers, M. and Wertz, V. (1990). *Adaptive Optimal Control. The Thinking Man's GPC*. Prentice-Hall, New York.
- Bittani, S.; Laub, A. J. and Willems, J. C., editors (1991). *The Riccati Equation*. Springer-Verlag, Berlin.
- Björck, Å. (1996). *Numerical Methods for Least Squares Problems*. SIAM, Philadelphia.
- Blanchini, F. (1999). Set Invariance in Control. *Automatica*, **35**, 1747–1767.

- Blot, J. and Hayek, N. (2001). Conjugate Points in Infinite-Horizon Control Problems. *Automatica*, **37**, 523–526.
- Bock, H. G. (1983). Recent Advances in Parameter Identification Techniques for O.D.E. In P. Deuffhard and E. Hairer, editors, *Numerical Treatment of Inverse Problems in Differential and Integral Equations*. Birkhäuser, Boston.
- Bock, H. G. and Plitt, K. J. (1983). A Multiple Shooting Algorithm for Direct Solution of Optimal Control Problems. In *9th IFAC World Congress Budapest*. Pergamon Press.
- Bock, H. G.; Diehl, M. M.; Leineweber, D. B. and Schlöder, J. P. (2000). A Direct Multiple Shooting Method for Real-Time Optimization of Nonlinear DAE Processes. In F. Allgöwer and A. Zheng, editors, *Nonlinear Model Predictive Control*, pages 245–267. Birkhäuser, Basel.
- Boggs, P. T. and Tolle, J. W. (1996). Sequential Quadratic Programming. *Acta Numerica*, **4**, 1–51.
- Boggs, P. T. and Tolle, J. W. (2000). Sequential Quadratic Programming for Large-Scale Nonlinear Optimization. *Journal of Computational and Applied Mathematics*, **124**, 123–137.
- Bonné, D.; Jørgensen, J. B.; Rawlings, J. B. and Jørgensen, S. B. (2003). Infinite Horizon Control Given Infeasible Targets. Manuscript in preparation.
- Bortolotto, G. and Jørgensen, S. B. (1986). Finite Horizon Multivariable Optimal Control of Constrained Chemical Processes. In *IFAC 6th World Congress*, London, England.
- Bortolotto, G. E. (1985). *Dynamics and Adaptive Control of a Fixed Bed Reactor with Recycle*. Ph.D. thesis, Technical University of Denmark.
- Box, G. E. P.; Jenkins, G. M. and Reinsel, G. C. (1994). *Time Series Analysis. Forecasting and Control*. Prentice-Hall, New Jersey, third edition.
- Brabrand, H. (1991). *Dynamics, Identification and Control of a Fixed Bed Reactor with Reactant Recycle*. Ph.D. thesis, Technical University of Denmark.
- Brock, W. A. and Haurie, A. (1976). On the Existence of Overtaking Optimal Trajectories over an Infinite-Time Horizon. *Mathematics of Operations Research*, **95**, 195–213.
- Bryson, A. E. (1996). Optimal Control - 1950 to 1985. *IEEE Control Systems*, pages 26–33.
- Bryson, A. E. and Ho, Y.-C. (1975). *Applied Optimal Control. Optimization, Estimation, and Control*. Taylor and Francis.

- Bulirsch, R. and Kraft, D., editors (1994). *Computational Optimal Control*. Birkhäuser, Basel.
- Camacho, E. F. and Bordons, C. (1999). *Model Predictive Control*. Springer, London.
- Canon, M. D.; Cullum, C. D. and Polak, E. (1970). *Theory of Optimal Control and Mathematical Programming*. McGraw-Hill, New York.
- Caracotsios, M. and Stewart, W. E. (1985). Sensitivity Analysis of Initial Value Problems with Mixed ODEs and Algebraic Equations. *Computers and Chemical Engineering*, **9**, 359–365.
- Carlson, D. A. (1986a). On the Existence of Catching-Up Optimal Solutions for Lagrange Problems Defined on Unbounded Intervals. *Journal of Optimization Theory and Applications*, **49**, 207–225.
- Carlson, D. A. (1986b). Existence of Finitely Optimal Solutions for Infinite-Horizon Optimal Control Problems. *Journal of Optimization Theory and Applications*, **51**, 41–62.
- Carlson, D. A. (1986c). A Carathéodory-Hamilton-Jacoby Theory for Infinite-Horizon Optimal Control Problems. *Journal of Optimization Theory and Applications*, **48**, 265–287.
- Carlson, D. A. (1987). On the Existence of Sporadically Catching-Up Optimal Solutions for Infinite-Horizon Optimal Control Problems. *Journal of Optimization Theory and Applications*, **53**, 219–235.
- Carlson, D. A. (1990). Uniformly Overtaking and Weakly Overtaking Optimal Solutions in Infinite-Horizon Optimal Control: When Optimal Solutions are Agreeable. *Journal of Optimization Theory and Applications*, **64**, 55–69.
- Carlson, D. A. and Haurie, A. (1987). *Infinite Horizon Optimal Control. Theory and Applications.*, volume 290 of *Lecture Notes in Economics and Mathematical Systems*. Springer-Verlag, Berlin.
- Carlson, D. A.; Haurie, A. and Jabrane, A. (1987). Existence of Overtaking Solutions to Infinite Dimensional Control Problems on Unbounded Intervals. *SIAM Journal of Control And Optimization*, **25**, 1517–1541.
- Carlson, D. A.; Haurie, A. B. and Leizarowitz, A. (1991). *Infinite Horizon Optimal Control, Deterministic and Stochastic Systems*. Lecture Notes in Economics and Mathematical Systems. Springer-Verlag, Berlin.
- Cervantes, A. M.; Wächter, A.; Tütüncü, R. H. and Biegler, L. T. (2000). A Reduced Space Interior Point Strategy for Optimization of Differential Algebraic Systems. *Computers and Chemical Engineering*, **24**, 39–51.

- Chamberlain, R. M. (1979). Some Examples of Cycling in Variable Metric Methods for Constrained Minimization. *Mathematical Programming*, **16**, 378–383.
- Chamberlain, R. M.; Lemaréchal, C.; Pedersen, H. C. and Powell, M. J. D. (1982). The Watchdog Technique for Forcing Convergence in Algorithms for Constrained Optimization. *Mathematical Programming Study*, **16**, 1–17.
- Chang, T. S. and Seborg, D. E. (1983). A Linear Programming Approach to Multivariable Feedback Control with Inequality Constraints. *International Journal of Control*, **37**, 583–597.
- Chen, C.-T. (1999). *Linear System Theory and Design*. Oxford, third edition.
- Chen, H. and Allgöwer, F. (1998). A Quasi-Infinite Horizon Nonlinear Model Predictive Control Scheme with Guaranteed Stability. *Automatica*, **34**, 1205–1217.
- Chen, H.-S. and Stadtherr, M. A. (1984). Enhancements of the Han-Powell Method for Successive Quadratic Programming. *Computers and Chemical Engineering*, **8**, 229–234.
- Chmielewski, D. and Manousiouthakis, V. (1996). On Constrained Infinite-Time Linear Quadratic Optimal Control. *Systems & Control Letters*, **29**, 121–129.
- Clarke, D., editor (1994). *Advances in Model-Based Predictive Control*. Oxford University Press, Oxford.
- Clarke, D. W. (1988). Application of Generalized Predictive Control to Industrial Processes. *IEEE Control Systems Magazine*, **122**, 49–55.
- Clarke, D. W.; Mohtadi, C. and Tuffs, P. S. (1987a). Generalized Predictive Control. Part 1: The Basic Algorithm. *Automatica*, **23**, 137–148.
- Clarke, D. W.; Mohtadi, C. and Tuffs, P. S. (1987b). Generalized Predictive Control. Part 2: Extensions and Interpretations. *Automatica*, **23**, 149–160.
- Cuthrell, J. E. and Biegler, L. T. (1987). On the Optimization of Differential-Algebraic Process Systems. *AIChE J.*, **33**, 1257–1270.
- Cuthrell, J. E. and Biegler, L. T. (1989). Simultaneous Optimization and Solution Methods for Batch Reactor Control Profiles. *Computers and Chemical Engineering*, **13**, 49–62.
- Cutler, C. R. and Ramaker, B. L. (1979). Dynamic Matrix Control - a Computer Control Algorithm. In *AIChE National Meeting*, Houston, TX.
- Cutler, C. R. and Ramaker, B. L. (1980). Dynamic Matrix Control - a Computer Control Algorithm. In *Joint Automatic Control Conference*. ACC.

- Cwikel, M. and Gutman, P.-O. (1986). Convergence of an Algorithm to Find Maximal State Constraint Sets for Discrete-Time Linear Dynamical Systems with Bounded Controls and States. *IEEE Trans. Automat. Contr.*, **31**, 457–459.
- De Keyser, R. M. C.; Van de Velde, P. G. A. and Dumortier, F. A. G. (1988). A Comparative Study of Self-Adaptive Long-Range Predictive Control Methods. *Automatica*, **24**, 149–163.
- De Nicolao, G.; Magni, L. and Scattolini, R. (2000). Stability and Robustness of Nonlinear Receding Horizon Control. In F. Allgöwer and A. Zheng, editors, *Nonlinear Model Predictive Control*, pages 3–22. Birkhäuser, Basel.
- de Oliveira, N. M. C. and Biegler, L. T. (1994). Constraint Handling and Stability Properties of Model-Predictive Control. *AIChE J.*, **40**, 1138–1155.
- Dennis, J. E. and Schnabel, R. B. (1996). *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Classics in Applied Mathematics. SIAM, Philadelphia.
- Dennis, J. E.; Gay, D. M. and Welsch, R. E. (1981). Algorithm 573 - NL2SOL, An Adaptive Nonlinear Least-Squares Algorithm. *ACM Transactions on Mathematical Software*, **7**, 348–368.
- Deuffhard, P. (1983). Order and Step-size Control in Extrapolation Methods. *Numerische Mathematik*, **41**, 399–422.
- Deuffhard, P.; Hairer, E. and Zugck, J. (1987). One-Step and Extrapolation Methods for Differential-Algebraic Systems. *Numerische Mathematik*, **51**, 501–516.
- Deuffhard, P.; Nowak, U. and Wulkow, M. (1990). Recent Developments in Chemical Computing. *Computers and Chemical Engineering*, **14**, 1249–1258.
- Deutsch, R. (1965). *Estimation Theory*. Prentice-Hall, Englewood Cliffs, NJ.
- Diehl, M.; Uslu, I.; Findeisen, R.; Schwarzkopf, S.; Allgöwer, F.; Bock, H. G.; Bürner, T.; Gilles, E. D.; Kienle, A.; Schlöder, J. P. and Stein, E. (2001). Real-Time Optimization for Large Scale Processes: Nonlinear Model Predictive Control of a High Purity Distillation Column. In M. Grötschel; S. O. Krumke and J. Rambau, editors, *Online Optimization of Large Scale Systems: State of the Art*. Springer, New York.
- Diehl, M.; Bock, H. G.; Schlöder, J. P.; Rolf Findeisen, Z. N. and Allgöwer, F. (2002). Real-Time Optimization and Nonlinear Model Predictive Control of Processes Governed by Differential-Algebraic Equations. *Journal of Process Control*, **12**, 577–585.
- Diehl, M. M. (2001). *Real-Time Optimization for Large Scale Nonlinear Processes*. Ph.D. thesis, Ruprecht-Karls-Universität Heidelberg.

- Dohrmann, C. R. and Robinett, R. D. (1997). Efficient Sequential Quadratic Programming Implementations for Equality-Constrained Discrete-Time Optimal Control. *Journal of Optimization Theory and Applications*, **95**, 323–346.
- Dollar, R.; Melton, L. L.; Morshedi, A. M.; Glasgow, D. T. and Repsher, K. W. (1993). Consider Adaptive Multivariable Predictive Controllers. *Hydrocarbon Processes*, pages 109–112.
- Draper, N. R. and Smith, H. (1998). *Applied Regression Analysis*. Wiley, New York.
- Dreyfus, S. E. (1964). Some Types of Optimal Control of Stochastic Systems. *SIAM Journal of Control*, **2**, 120–134.
- Duff, I. S.; Erisman, A. M. and Reid, J. K. (1986). *Direct Methods for Sparse Matrices*. Oxford Science Publications, Oxford.
- Dunn, J. C. and Bertsekas, D. P. (1989). Efficient Dynamic Programming Implementations of Newton's Method for Unconstrained Optimal Control Problems. *Journal of Optimization Theory and Applications*, **63**, 23–38.
- Dyer, P. and McReynolds, S. R. (1970). *The Computation and Theory of Optimal Control*. Academic Press, New York.
- Eaton, J. W. and Rawlings, J. B. (1992). Model-Predictive Control of Chemical Processes. *Chem. Eng. Sci.*, **47**, 705–720.
- Eldersveld, S. K. (1991). *Large-Scale Sequential Quadratic Programming Algorithms*. Ph.D. thesis, Department of Operations Research, Stanford University.
- Everett, H. (1963). Generalized Lagrange Multiplier Method for Solving Problems of Optimum Allocation of Resources. *Operations Research*, **2**, 399–417.
- Fan, L. T. and Wang, C. S. (1964). *The Discrete Maximum Principle*. Wiley.
- Feehery, W. F. and Barton, P. I. (1998). Dynamic Optimization with State Variable Path Constraints. *Computers and Chemical Engineering*, **22**, 1241–1256.
- Feehery, W. F.; Tolsma, J. E. and Barton, P. I. (1997). Efficient Sensitivity Analysis of Large-Scale Differential-Algebraic Equations. *Appl. Numer. Math.*, **25**, 41–54.
- Feinstein, C. D. and Luenberger, D. G. (1981). Analysis of the Asymptotic Behavior of Optimal Control Trajectories: The Implicit Programming Problem. *SIAM Journal of Control and Optimization*, **19**, 561–585.



- Feldbaum, A. A. (1961a). Dual Control Theory. *Automation and Remote Control*, **21**, 874–880.
- Feldbaum, A. A. (1961b). Dual Control Theory. II. *Automation and Remote Control*, **21**, 1033–1039.
- Feldbaum, A. A. (1965). *Optimal Control Systems*. Academic Press, New York.
- Fikar, M. and Jørgensen, S. B. (1994). Multivariable Constrained Control Based Upon All Stabilizing Controller. In D. Clarke, editor, *Advances in Model-Based Predictive Control*, pages 260–275. Oxford Science Publications, Oxford.
- Fletcher, R. (1987). *Practical Methods of Optimization*. Wiley, New York.
- Fletcher, R. and Xu, C. (1987). Hybrid Methods for Nonlinear Least Squares. *IMA Journal of Numerical Analysis*, **7**, 371–389.
- Francis, B. A. and Wonham, W. M. (1976). The Internal Model Principle of Control Theory. *Automatica*, **12**, 457–465.
- Franklin, G. F.; Powell, J. D. and Workman, M. (1998). *Digital Control of Dynamic Systems*. Addison Wesley, Menlo Park, CA, third edition.
- Froisy, J. B. (1994). Model Predictive Control: Past, Present and Future. *ISA Trans.*, **33**, 235–243.
- Gajic, Z. and Qureshi, M. T. J. (1995). *Lyapunov Matrix Equation in System Stability and Control*, volume 195 of *Mathematics in Science and Engineering*. Academic Press, San Diego, California.
- Galán, S.; Feehery, W. F. and Barton, P. I. (1999). Parametric Sensitivity Functions for Hybrid Discrete/Continuous Systems. *Applied Numerical Mathematics*, **31**, 17–47.
- Garcia, C. E. and Morari, M. (1981). Optimal Operation of Integrated Processing Systems, Part I: Open-Loop On-Line Optimizing Control. *AIChE J.*, **27**, 960–968.
- Garcia, C. E. and Morari, M. (1984). Optimal Operation of Integrated Processing Systems, Part II: Closed-Loop On-Line Optimizing Control. *AIChE J.*, **30**, 226–234.
- Garcia, C. E. and Morshedi, A. M. (1986). Quadratic Programming Solution of Dynamic Matrix Control (QDMC). *Chemical Engineering Communications*, **46**, 73–87.
- Garcia, C. E.; Prett, D. M. and Morari, M. (1989). Model Predictive Control: Theory and Practice - A Survey. *Automatica*, **25**, 335–348.

- Garcia-Palomares, U. M. and Mangasarian, O. L. (1976). Superlinearly Convergent Quasi-Newton Algorithms for Nonlinearly Constrained Optimization Problems. *Mathematical Programming*, **11**, 1–13.
- Gertz, M. and Wright, S. (2001a). Object-Oriented Software for Quadratic Programming. Technical Report Preprint ANL/MCS-P891-1000, Mathematics and Computer Science Division, Argonne National Laboratory.
- Gertz, M. and Wright, S. J. (2001b). *OOQP - User Guide*. Department of Computer Science, University of Wisconsin-Madison.
- Gevers, M. (2003). A Personal View on the Development of System Identification. In *13th IFAC Symposium on System Identification, 27-29 August 2003, Rotterdam, The Netherlands*. IFAC.
- Gevers, M. (2004). Identification for Control: Achievements and Open Problems. In *DYCOPS 7, Cambridge, Massachusetts USA, July 5-7*. IFAC.
- Gilbert, E. G. and Tan, K. T. (1991). Linear Systems with State and Control Constraints: The Theory and Application of Maximal Output Admissible Sets. *IEEE Trans. Automat. Contr.*, **36**, 1008–1020.
- Gill, P. E. and Murray, W. (1978). Numerically Stable Methods for Quadratic Programming. *Mathematical Programming*, **14**, 349–372.
- Gill, P. E.; Murray, W.; Saunders, M. A. and Wright, M. H. (1991). Inertia-Controlling Methods for General Quadratic Programming. *SIAM Review*, **33**, 1–36.
- Gill, P. E.; Murray, W.; Saunders, M. A. and Wright, M. H. (1992). Some Theoretical Properties of an Augmented Lagrangian Merit Function. In P. M. Pardalos, editor, *Advances in Optimization and Parallel Computing*, pages 101–128. Elsevier Science Publishers.
- Gill, P. E.; Murray, W. and Saunders, M. E. (1995). User's Guide for QPOPT 1.0: A Fortran Package for Quadratic Programming. Technical Report SOL 95-4, Systems Optimization Laboratory, Department of Operations Research, Stanford University.
- Gill, P. E.; Murray, W. and Saunders, M. A. (1997). SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization. Technical Report SOL 97-3, Stanford University.
- Goldfarb, D. and Idnani, A. (1983). A Numerically Stable Dual Method for Solving Strictly Convex Quadratic Programs. *Mathematical Programming*, **27**, 1–33.
- Golub, G. H. and Van Loan, C. F. (1996). *Matrix Computations*. Johns Hopkins, third edition.

- Goodwin, G. C. (2002). Inverse Problems with Constraints. In *15th Triennial World Congress*, Barcelona, Spain.
- Goodwin, G. C. and Sin, K. S. (1984). *Adaptive Filtering, Prediction and Control*. Prentice-Hall, New Jersey.
- Gopaluni, R. B.; Patwardhan, R. S. and Shah, S. L. (2003). The Nature of Data Pre-Filters in MPC Relevant Identification - Open- and Closed-Loop Issues. *Automatica*, **39**, 1617–1626.
- Gopaluni, R. B.; Patwardhan, R. S. and Shah, S. L. (2004). MPC Relevant Identification - Tuning the Noise Model. *Journal of Process Control*, **14**, 699–714.
- Greco, C.; Menga, G.; Mosca, E. and Zappa, G. (1984). Performance Improvements of Self Tuning Controllers by Multistep Horizons: The MUSMAR Approach. *Automatica*, **20**, 681–700.
- Griewank, A. and Toint, P. L. (1982). Partitioned Variable Metric Updates for Large Structured Optimization Problems. *Numer. Math.*, **39**, 119–137.
- Gropp, W.; Lusk, E. and Skjellum, A. (1994). *Using MPI. Portable Parallel Programming with the Message-Passing Interface*. The MIT Press, Cambridge, Massachusetts.
- Grosdidier, P.; Froisy, B. and Hammann, M. (1988). The IDCOM-M Controller. In T. J. McAvoy; Y. Arkun and E. Zafiriou, editors, *Proceedings of the 1988 IFAC Workshop on Model Based Control*, pages 31–36, Oxford. Pergamon Press.
- Grötschel, M.; Krumke, S. O. and Rambau, J., editors (2001). *Online Optimization of Large Scale Systems*. Springer Verlag.
- Gunckel, T. F. and Franklin, G. F. (1963). A General Solution for Linear Sampled Data Control. *Trans ASME, J. Basic Eng.*, **82-D**, 35–45.
- Gutman, P.-O. (1982). *Controllers for Bilinear and Constrained Linear Systems*. Ph.D. thesis, Department of Automatic Control, Lund Institute of Technology.
- Gutman, P.-O. and Cwikel, M. (1986). Admissible Sets and Feedback Control for Discrete-Time Linear Dynamical Systems with Bounded Controls and States. *IEEE Trans. Automat. Contr.*, **31**, 373–376.
- Gutman, P.-O. and Cwikel, M. (1987). An Algorithm to Find Maximal State Constraint Sets for Discrete-Time Linear Dynamical Systems with Bounded Controls and States. *IEEE Trans. Automat. Contr.*, **32**, 251–254.

- Haber, R.; Schmitz, U. and Bars, R. (2003). Long-Range Optimal Model and Multi-Step-Ahead Prediction Identification for Predictive Control. In *13th IFAC Symposium on Identification*, Rotterdam, The Netherlands.
- Hairer, E. and Wanner, G. (1996). *Solving Ordinary Differential Equations II. Stiff and Differential-Algebraic Problems*. Springer-Verlag, second edition.
- Hairer, E.; Nørsett, S. P. and Wanner, G. (1993). *Solving Ordinary Differential Equations I. Nonstiff Problems*. Springer-Verlag, second edition.
- Halkin, H. (1966). A Maximum Principle of the Pontryagin Type for Systems Described by Nonlinear Difference Equations. *SIAM Journal of Control*, **4**, 90–111.
- Halkin, H. (1974). Necessary Conditions for Optimal Control Problems with Infinite Horizons. *Econometrica*, **42**, 267–272.
- Hallager, L. (1984). *Multivariable Self-Tuning Control of a Chemical Fixed-Bed Reactor Using Structured Linear Models*. Ph.D. thesis, Technical University of Denmark.
- Hallager, L.; Goldschmidt, L. and Jørgensen, S. B. (1984). Multivariable Adaptive Identification and Control of a Distributed Chemical Reactor. *Large Scale Systems*, **6**, 323–336.
- Hamilton, J. D. (1994). *Time Series Analysis*. Princeton University Press, Princeton, New Jersey.
- Hammond, P. J. and Kennan, J. (1979). Uniformly Optimal Infinite-Horizon Plans. *International Economic Review*, **3**, 283–296.
- Hammond, P. J. and Mirrless, J. A. (1973). Agreeable Plans. In J. A. Mirrless and N. H. Stern, editors, *Models of Economic Growth*, pages 283–299. Macmillan, New York.
- Han, S. P. (1976). Superlinearly Convergent Variable-Metric Algorithms for General Nonlinear Programming Problems. *Mathematical Programming*, **11**, 263–282.
- Han, S. P. (1977). A Globally Convergent Method for Nonlinear Programming. *Journal of Optimization Theory and Applications*, **22**, 297–310.
- Hansen, P. C. (1996). *Rank-Deficient and Discrete Ill-Pose Problems*. Polyteknisk Forlag, Lyngby, Denmark.
- Hansen, P. C. (1997). *Rank-Deficient and Discrete-Ill Posed Problems: Numerical Aspects of Linear Inversion*. SIAM, Philadelphia.
- Hartl, R. F.; Sethi, S. P. and Vickson, R. G. (1995). A Survey of the Maximum Principles for Optimal Control Problems with State Constraints. *SIAM Review*, **37**, 181–218.

- Hassibi, B.; Sayed, A. H. and Kailath, T. (1999). *Indefinite-Quadratic Estimation and Control. A Unified Approach to  $H^2$  and  $H^\infty$  Theories*. SIAM, Philadelphia.
- Hastie, T.; Tibshirani, R. and Friedman, J. (2001). *The Elements of Statistical Learning. Data Mining, Inference and Prediction*. Springer, New York.
- Haurie, A. (1976). Optimal Control on the Infinite Horizon: The Turnpike Approach. *Journal of Mathematical Economics*, **3**, 81–102.
- Haurie, A. (1980). Existence and Global Asymptotic Stability of Optimal Trajectories for a Class of Infinite-Horizon Nonconvex Systems. *Journal of Optimization Theory and Applications*, **31**, 515–533.
- Haurie, A. and Sethi, S. (1984). Decision and Forecast Horizons, Agreeable Plans, and the Maximum Principle for Infinite-Horizon Control Problems. *Operations Research Letters*, **3**, 261–265.
- Heath, J. A. (1996). *Process Control Structure Selection Based on Optimization*. Ph.D. thesis, Department of Chemical Engineering and Chemical Technology, Imperial College of Science, Technology and Medicine.
- Henson, M. A. and Seborg, D. E. (1997). *Nonlinear Process Control*. Prentice Hall, Upper Saddle River, New Jersey.
- Hindmarsh, A. C. and Serban, R. (2002). User Documentation for CVODES, An ODE Solver with Sensitivity Analysis Capabilities. Technical Report UCRL-MA-148813, Center for Applied Scientific Computing, Lawrence Livermore National Laboratory.
- Hjalmarsson, H. (2003). From Experiments to Closed Loop Control. In *13th IFAC Symposium on System Identification, 27-29 August 2003, Rotterdam, The Netherlands*. IFAC.
- Holtzman, J. M. (1966a). Convexity and the Maximum Principle for Discrete Systems. *IEEE Transactions on Automatic Control*, **AC-11**, 30–35.
- Holtzman, J. M. (1966b). On the Maximum Principle for Nonlinear Discrete-Time Systems. *IEEE Transactions on Automatic Control*, **AC-11**, 273–274.
- Holtzman, J. M. and Halkin, H. (1966). Directional Convexity and the Maximum Principle for Discrete Systems. *SIAM Journal of Control*, **4**, 263–275.
- Horn, F. and Jackson, R. (1965). Discrete Maximum Principle. *Industrial and Engineering Chemistry Fundamentals*, **4**, 487–488.
- Hoza, M. and Stadtherr, M. A. (1993). An Improved Watchdog Line Search for Successive Quadratic Programming. *Computers and Chemical Engineering*, **17**, 943–947.

- Jackson, R. and Horn, F. (1965). On Discrete Analogues of Pontryagin's Maximum Principle. *International Journal of Control*, **1**, 389–395.
- Jacobson, D. H. and Mayne, D. Q. (1970). *Differential Dynamic Programming*. American Elsevier, New York.
- Jazwinski, A. H. (1970). *Stochastic Processes and Filtering Theory*. Academic Press.
- Jonson, H. (1983). *A Newton Method for Solving Non-Linear Optimal Control Problems with General Constraints*. Ph.D. thesis, Linköping University.
- Jørgensen, J. B. (2004a). Dual Active Set Algorithm for Convex Quadratic Programming. *submitted to SIAM Journal on Optimization*.
- Jørgensen, J. B. (2004b). A Fast Feedforward-Feedback Predictive Controller. *Manuscript in preparation*.
- Jørgensen, J. B. (2004c). *Quadratic Programming*. Manuscript for Research Monography.
- Jørgensen, J. B. (2004d). An Implementation of the Dual Active Set QP Algorithm. *Submitted to SIAM Journal on Optimization*.
- Jørgensen, J. B. (2004e). Schur Complement Factorization for the Dual Active Set QP Algorithm. *SIAM Journal on Optimization and Control (to be submitted)*.
- Jørgensen, J. B. and Jørgensen, S. B. (2000). Towards Automatic Decentralized Control Structure Selection. *Computers and Chemical Engineering*, **24**, 841–846.
- Jørgensen, J. B. and Rawlings, J. B. (2000). Tracking and Infeasibilities in Model Predictive Control. Seminar presentation at Department of Automatic Control, Lund Institute of Technology.
- Jørgensen, J. B.; Rawlings, J. B. and Jørgensen, S. B. (2002). Computational Principles of Model Predictive Control. *Manuscript in Preparation*.
- Jørgensen, J. B.; Rawlings, J. B. and Jørgensen, S. B. (2004). Numerical Methods for Large Scale Moving Horizon Estimation and Control. In *DY-COPS 7, Cambridge, MA, July 5-7*.
- Jørgensen, S. B. and Lee, J. H. (2001). Recent Advances and Challenges in Process Identification. In J. B. Rawlings and B. A. Ogunnaike, editors, *Chemical Process Control - 6: Assessment and New Directions for Research*.
- Joseph, B. and Brosilow, C. (1978). Construction of Optimal and Suboptimal Estimators. *AIChE J.*, **24**, 500–509.

- Joseph, P. D. and Tou, J. T. (1961). On Linear Control Theory. *Trans AIEE, Part III*, **80**, 193–196.
- Kailath, T. (1975). Supplement to "A Survey of Data Smoothing". *Automatica*, **11**, 109–111.
- Kailath, T.; Sayed, A. H. and Hassibi, B. (2000). *Linear Estimation*. Prentice Hall.
- Kalman, R. E. (1960a). Contributions to the Theory of Optimal Control. *Boletín de la Sociedad Matematica Mexicana*, **5**, 102–119.
- Kalman, R. E. (1960b). A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering Transactions of the ASME, Series D*, **82**, 35–45.
- Kalman, R. E. and Bertram, J. E. (1960). Control System Analysis and Design Via the "Second Method" of Lyapunov. II. Discrete-Time Systems. *J. Basic Engr.*, **82 D**, 394–400.
- Kaminski, P. G.; Bryson, A. E. and Schmidt, S. F. (1971). Discrete Square-Root Filtering: A Survey of Current Techniques. *IEEE Transactions on Automatic Control*, **16**, 727–736.
- Kantor, J. C.; Garcia, C. E. and Carnahan, B., editors (1997). *Chemical Process Control-V. Assessment and New Directions for Research*, Tahoe City, California. CACHE, American Institute of Chemical Engineers.
- Kassmann, D. A.; Badgwell, T. A. and Hawkins, R. B. (2000). Robust Steady-State Target Calculation for Model Predictive Control. *AIChE Journal*, **46**, 1007–1024.
- Katz, S. (1962a). A Discrete Version of Pontryagin's Maximum Principle. *Journal of Electronics and Control*, **13**, 179.
- Katz, S. (1962b). Best Operating Points for Staged Systems. *Industrial and Engineering Chemistry Fundamentals*, **1**.
- Keerthi, S. S. (1986). *Optimal Feedback Control of Discrete-Time Systems with State-Control Constraints and General Cost Functions*. Ph.D. thesis, University of Michigan.
- Keerthi, S. S. and Gilbert, E. G. (1987). Computation of Minimum-Time Feedback Control Laws for Discrete-Time Systems with State-Control Constraints. *IEEE Trans. Automat. Contr.*, **32**, 432–435.
- Keerthi, S. S. and Gilbert, E. G. (1988). Optimal Infinite-Horizon Feedback Laws for a General Class of Constrained Discrete-Time Systems: Stability and Moving Horizon Approximations. *Journal of Optimization Theory and Applications*, **57**, 265–293.

- Kerrigan, E. C. and Maciejowski, J. M. (2000). Soft Constraints and Exact Penalty Functions in Model Predictive Control. In *Control 2000*.
- Kiehl, M. (1999). Sensitivity Analysis of ODEs and DAEs - Theory and Implementation Guide. *Optimization Methods and Software*, **10**, 803–821.
- Knudsen, J. K. H. (1975). Time-Optimal Computer Control of a Pilot Plant Evaporator. In *IFAC 6th World Congress*, volume Part 2, pages 45.3/1–6.
- Koggersbøl, A. (1995). *Distillation Column Dynamics, Operability and Control*. Ph.D. thesis, Department of Chemical Engineering, Technical University of Denmark.
- Koung, C.-V. and MacGregor, J. F. (1994). Identification for Robust Multivariable Control: The Design of Experiments. *Automatica*, **30**, 1541–1554.
- Kraft, D. (1985). On Converting Optimal Control Problems Into Nonlinear Programming Problems. In K. Schittkowski, editor, *Computational Mathematical Programming*, NATO ASI Series, pages 261–280. Springer.
- Kraft, D. (1994). Algorithm 733: TOMP - Fortran Modules for Optimal Control Calculations. *ACM Transactions on Mathematical Software*, **20**, 262–281.
- Kristensen, M. R.; Jørgensen, J. B.; Thomsen, P. G. and Jørgensen, S. B. (2004a). An ESDIRK Method with Sensitivity Analysis Capabilities. *Computers and Chemical Engineering*, **28**, 2695–2707.
- Kristensen, M. R.; Jørgensen, J. B.; Thomsen, P. G. and Jørgensen, S. B. (2004b). Efficient Sensitivity Computation for Nonlinear Model Predictive Control. In F. Allgöwer, editor, *NOLCOS 2004, 6th IFAC-Symposium on Nonlinear Control Systems, September 01-04, 2004, Stuttgart, Germany*, pages 723–728. IFAC.
- Kristensen, M. R.; Jørgensen, J. B.; Thomsen, P. G.; Michelsen, M. and Jørgensen, S. B. (2005). Sensitivity Analysis in Index-1 Differential Algebraic Equations by ESDIRK Methods. In *Accepted for Publication in IFAC Annual World Congress 2005*.
- Kristensen, N. R. (2002). *Fed-Batch Process Modelling for State Estimation and Optimal Control*. Ph.D. thesis, Technical University of Denmark.
- Kristensen, N. R.; Melgaard, H. and Madsen, H. (2002). *CTSM 2.1. User's Guide*. Technical University of Denmark.
- Kristensen, N. R.; Madsen, H. and Jørgensen, S. B. (2004c). Parameter Estimation in Stochastic Grey-Box Models. *Automatica*, **40**, 225–237.
- Kwakernaak, H. and Sivan, R. (1972). *Linear Optimal Control Systems*. Wiley.



- Larimore, W. E. (1983). System Identification, Reduced-Order Filtering and Modeling Via Canonical Variate Analysis. In H. S. Rao and T. Dorato, editors, *Proc. 1983 American Control Conference*, pages 445–451, New York. IEEE.
- LaSalle, J. P. (1976). *The Stability of Dynamical Systems*. SIAM, Philadelphia, Pennsylvania.
- LaSalle, J. P. (1986). *The Stability and Control of Discrete Processes*. Applied Mathematical Sciences 62. Springer-Verlag, New York.
- Lee, E. B. and Markus, L. (1968). *Foundations of Optimal Control Theory*. Wiley, New York.
- Lee, J. H. (2000). Modeling and Identification for Nonlinear Model Predictive Control: Requirements, Current Status and Future Research Needs. In F. Allgöwer and A. Zheng, editors, *Nonlinear Model Predictive Control*, pages 269–293. Birkhäuser, Basel.
- Lee, J. H. and Cooley, B. (1997). Recent Advances in Model Predictive Control and Other Related Areas. In J. C. Kantor; C. E. Garcia and B. Carnahan, editors, *Chemical Process Control-V*, volume 93 of *AIChE Symposium Series No. 316*, pages 201–216. CACHE, American Institute of Chemical Engineers.
- Lee, J. H. and Cooley, B. L. (1998). Optimal Feedback Control Strategies for State-Space Systems with Stochastic Parameters. *IEEE Transactions on Automatic Control*, **43**, 1469–1475.
- Lee, J. H.; Gelormino, M. S. and Morari, M. (1992). Model Predictive Control of Multi-Rate Sampled-Data Systems: A State Space Approach. *International Journal of Control*, **55**, 153.
- Lee, J. H.; Morari, M. and Garcia, C. E. (1994). State-Space Interpretation of Model Predictive Control. *Automatica*, **30**, 707–717.
- Lee, S. L.; Hindmarsh, A. C. and Brown, P. N. (2000). User Documentation for SensPVODE, a Variant of PVODE for Sensitivity Analysis. Technical Report UCRL-MA-140211, Lawrence Livermore National Laboratory.
- Leineweber, D. B. (1995). *The Theory of MUSCOD in a Nutshell*. Master’s thesis, Interdisziplinäres Zentrum für Wissenschaftliches Rechnen (IWR), Universität Heidelberg.
- Leineweber, D. B. (1999). *Efficient Reduced SQP Methods for the Optimization of Chemical Processes Described by Large Sparse DAE Models*, volume 613 of *Fortschritt Berichte. VDI Reihe 3, Verfahrenstechnik*. VDI Verlag, Düsseldorf.

- Leineweber, D. B.; Bauer, I.; Bock, H. G. and Schlöder, J. P. (2001). An Efficient Multiple Shooting Based Reduced SQP Strategy for Large-Scale Dynamic Process Optimization. Technical Report IWR-2001-23, Interdisciplinary Center for Scientific Computing, University of Heidelberg.
- Leis, J. R. and Kramer, M. A. (1985). Sensitivity Analysis of Systems of Differential and Algebraic Equations. *Computers and Chemical Engineering*, **9**, 93–96.
- Leizarowitz, A. (1987a). Infinite Horizon Optimization for Finite State Markov Chains. *SIAM Journal of Control and Optimization*, **25**, 1601–1618.
- Leizarowitz, A. (1987b). Infinite Horizon Stochastic Regulation and Tracking with the Overtaking Criterion. *Stochastics*, **22**, 117–150.
- Li, S. and Petzold, L. R. (1999). Design of New DASPK for Sensitivity Analysis. Technical report, UCSB Department of Computer Science.
- Li, S. and Petzold, L. R. (2001). Software and Algorithms for Sensitivity Analysis of Large-Scale Differential-Algebraic Systems. *J. Comp. and Appl. Math.*.
- Li, S.; Lim, K. Y. and Fisher, D. G. (1989). A State Space Formulation for Model Predictive Control. *AIChE J.*, **35**, 241–249.
- Liao, L.-Z. and Shoemaker, C. A. (1991). Convergence in Unconstrained Discrete-Time Differential Dynamic Programming. *IEEE Transactions on Automatic Control*, **36**, 692–706.
- Ljung, L. (1999). *System Identification. Theory for the User*. Prentice Hall, Upper Saddle River, NJ, second edition.
- Loeblein, C. and Perkins, J. D. (1999a). Structural Design for On-Line Process Optimization: I. Dynamic Economics of MPC. *AIChE J.*, **45**, 1018–1029.
- Loeblein, C. and Perkins, J. D. (1999b). Structural Design for On-Line Process Optimization: II. Application to a Simulated FCC. *AIChE J.*, **45**, 1030–1040.
- Logsdon, J. S. and Biegler, L. T. (1993). A Relaxed Reduced Space SQP Strategy for Dynamic Optimization Problems. *Computers and Chemical Engineering*, **17**, 367–372.
- Luenberger, D. G. (1969). *Optimization by Vector Space Methods*. Wiley.
- Luenberger, D. G. (1972). Mathematical Programming and Control Theory: Trends of Interplay. In A. D. Geoffrion, editor, *Perspectives on Optimization: A Collection of Expository Articles*, pages 102–133. Addison-Wesley.
- Lütkepohl, H. (1993). *Introduction to Multiple Time Series Analysis*. Springer-Verlag, Berlin, second edition.

- Maciejowski, J. M. (2002). *Predictive Control with Constraints*. Prentice Hall, Harlow, England.
- Madsen, H. (1995). *Tidsrækkenalyse*. IMM, Lyngby, second edition.
- Malinvaud, E. (1969). First Order Certainty Equivalence. *Econometrica*, **37**, 706–718.
- Maly, T. and Petzold, L. (1996). Numerical Methods and Software for Sensitivity Analysis of Differential-Algebraic Systems. *Applied Numerical Mathematics*, **20**, 57–79.
- Mangasarian, O. L. (1994). *Nonlinear Programming*. Classics in Applied Mathematics. SIAM, Philadelphia.
- Maratros, N. (1978). *Exact Penalty Function Algorithms for Finite Dimensional and Control Optimization Problems*. Ph.D. thesis, University of London.
- Marquis, P. and Broustail, J. P. (1988). SMOC, a Bridge Between State Space and Model Predictive Controllers: Application to the Automation of a Hydrotreating Unit. In T. J. McAvoy; Y. Arkun and E. Zafiriou, editors, *Proceedings of the 1988 IFAC Workshop on Model Based Process Control*, pages 37–43, Oxford. Pergamon Press.
- Martin-Sanchez, J. M. and Rodellar, J. (1996). *Adaptive Predictive Control. From the Concepts to Plant Optimization*. Prentice-Hall.
- Martinsen, F.; Biegler, L. T. and Foss, B. A. (2002). Application of Optimization Algorithms to Nonlinear MPC. In *15th Triennial World Congress of the International Federation of Automatic Control*, Barcelona, Spain.
- Mayne, D. (1966). A Second Order Gradient Method for Determining Optimal Trajectories for Nonlinear Discrete-Time Systems. *International Journal of Control*, **3**, 85–95.
- Mayne, D. Q. (1995). Optimization in Model Based Control. In J. B. Rawlings, editor, *Proceedings of the IFAC Symposium on Dynamics and Control of Chemical Reactors and Batch Processes (DYCORD+’95)*, pages 229–242, Oxford. Elsevier Science.
- Mayne, D. Q. (1997). Nonlinear Model Predictive Control: An Assessment. In J. C. Kantor; C. E. Garcia and B. Carnahan, editors, *Chemical Process Control-V. Assessment and New Directions for Research*, pages 217–231, Tahoe City, California. CACHE, American Institute of Chemical Engineers.
- Mayne, D. Q. and Michalska, H. (1990). Receding Horizon Control of Nonlinear Systems. *IEEE Trans. Automat. Contr.*, **35**, 814–824.

- Mayne, D. Q.; Rawlings, J. B.; Rao, C. V. and Scokaert, P. O. M. (2000). Constrained Model Predictive Control: Stability and Optimality. *Automatica*, **36**, 789–814.
- Meditch, J. S. (1973). A Survey of Data Smoothing for Linear and Nonlinear Systems. *Automatica*, **9**, 151–162.
- Mehrotra, S. (1992). On the Implementation of a Primal-Dual Interior Point Method. *SIAM J. Optimization*, **2**, 575–601.
- Mehrotra, S. (1993). Quadratic Convergence in a Primal-Dual Method. *Mathematics of Operations Research*, **18**, 741–751.
- Melgaard, H. (1994). *Identification of Physical Models*. Ph.D. thesis, Technical University of Denmark.
- Mitter, S. K. (1966). Successive Approximation Methods for the Solution of Optimal Control Problems. *Automatica*, **3**, 135–149.
- Moler, C. and Van Loan, C. (1978). Nineteen Dubious Ways to Compute the Exponential of a Matrix. *SIAM Review*, **20**, 801–836.
- Morari, M. (1983). Design of Resilient Processing Plants - III. A General Framework for the Assessment of Dynamic Resilience. *Chem. Eng. Sci.*, **38**, 1881–1891.
- Morari, M. and Lee, J. H. (1999). Model Predictive Control: Past, Present and Future. *Computers and Chemical Engineering*, **23**, 667–682.
- Morari, M. and Stephanopoulos, G. (1980). Optimal Selection of Secondary Measurements Within the Framework of State Estimation in the Presence of Persistent Unknown Disturbances. *AIChE J.*, **26**, 247–260.
- Morari, M. and Zafiriou, E. (1989). *Robust Process Control*. Prentice-Hall, Englewood Cliffs, New Jersey.
- Morf, M. and Kailath, T. (1975). Square Root Algorithms for Least Squares Estimation. *IEEE Transactions on Automatic Control*, **20**, 487–497.
- Morshedi, A. M.; Cutler, C. R. and Skrovanek, T. A. (1985). Optimal Solution of Dynamic Matrix Control with Linear Programming Techniques (LDMC). In *Americical Control Conference*.
- Mosca, E. (1995). *Optimal, Predictive, and Adaptive Control*. Prentice-Hall.
- Mosca, E. and Casavola, A. (1995). Deterministic LQ Preview Tracking Design. *IEEE Trans. Automat. Contr.*, **40**, 1278–1281.
- Murray, D. M. and Yakowitz, S. J. (1979). Constrained Differential Dynamic Programming and Its Application to Multireservoir Control. *Water Resources Research*, **15**, 1017–1027.

- Murray, D. M. and Yakowitz, S. J. (1981). The Application of Optimal Control Methodology to Nonlinear Programming Problems. *Mathematical Programming*, **21**, 331–347.
- Murray, D. M. and Yakowitz, S. J. (1984). Differential Dynamic Programming and Newton’s Method for Discrete Optimal Control Problems. *Journal of Optimization Theory and Applications*, **43**, 395–414.
- Muske, K. R. (1995). *Linear Model Predictive Control of Chemical Processes*. Ph.D. thesis, The University of Texas at Austin.
- Muske, K. R. (1997). Steady-State Target Optimization in Linear Model Predictive Control. In *Proceedings of the American Control Conference*, pages 3597–3601, Albuquerque, New Mexico.
- Muske, K. R. and Badgwell, T. A. (2002). Disturbance Modeling for Offset-Free Linear Model Predictive Control. *Journal of Process Control*, **12**, 617–632.
- Muske, K. R. and Rawlings, J. B. (1993a). Model Predictive Control with Linear Models. *AIChE J.*, **39**, 262–287.
- Muske, K. R. and Rawlings, J. B. (1993b). Linear Model Predictive Control of Unstable Processes. *Journal of Process Control*, **3**, 85–96.
- Nahorski, Z.; Ravn, H. F. and Vidal, R. V. V. (1984). The Discrete-Time Maximum Principle: A Survey and Some New Results. *International Journal of Control*, **40**, 533–554.
- Narraway, L. T.; Perkins, J. D. and Barton, G. W. (1991). Interaction Between Process Design and Process Control: Economic Analysis of Process Dynamics. *J. Proc. Cont.*, **1**, 243.
- Nielsen, C. S. (1990). *Multivariable Identification and Control of an Experimental Distillation Column with Heat Pump*. Ph.D. thesis, Department of Chemical Engineering, Technical University of Denmark.
- Nielsen, J. N.; Madsen, H. and Young, P. C. (2000). Parameter Estimation in Stochastic Differential Equations: An Overview. *Annual Reviews in Control*, **24**, 83–94.
- Nocedal, J. and Wright, S. J. (1999). *Numerical Optimization*. Springer-Verlag, New York.
- Nørgaard, M.; Ravn, O.; Poulsen, N. K. and Hansen, L. K. (2000). *Neural Networks for Modelling and Control of Dynamic Systems*. Springer, London.
- Odelson, B. J. (2003). *Estimating Disturbance Covariances from Data for Improved Control Performance*. Ph.D. thesis, University of Wisconsin-Madison.

- Ogunnaike, B. A. and Ray, W. H. (1994). *Process Dynamics, Modeling, and Control*. Oxford University Press, New York.
- Ohno, K. (1978a). A New Approach to Differential Dynamic Programming for Discrete Time Systems. *IEEE Transactions on Automatic Control*, **AC-23**, 37–47.
- Ohno, K. (1978b). Differential Dynamic Programming and Separable Programs. *Journal of Optimization Theory and Applications*, **24**, 617–637.
- Ouellette, D. V. (1981). Schur Complements and Statistics. *Linear Algebra and Its Applications*, **36**, 187–295.
- Pannocchia, G. and Rawlings, J. B. (2001). The Velocity Algorithm LQR: A Survey. Technical Report TWMCC 2001-01, Department of Chemical Engineering, University of Wisconsin-Madison.
- Pannocchia, G. and Rawlings, J. B. (2003). Disturbance Models for Offset-Free Model-Predictive Control. *AIChE J.*, **49**, 426–437.
- Pannocchia, G.; Wright, S. J. and Rawlings, J. B. (2001). Model Predictive Control with Active Steady-State Constraints: Existence and Computation. Technical Report TWMCC 2001-04, Texas-Wisconsin Modeling and Control Consortium.
- Pannocchia, G.; Wright, S. J. and Rawlings, J. B. (2002). Existence and Computation of Infinite Horizon Model Predictive Control with Active Steady-State Input Constraints. *submitted to IEEE Transactions on Automatic Control*.
- Pantoja, J. F. A. D. (1988). Differential Dynamic Programming and Newton's Method. *International Journal of Control*, **47**, 1539–1553.
- Pantoja, J. F. A. D. and Mayne, D. Q. (1989). Sequential Quadratic Programming Algorithm for Discrete Optimal Control Problems with Control Inequality Constraints. *Int. J. Control*, **53**, 823–836.
- Pesch, H. J. (1994). A Practical Guide to the Solution of Real-Life Optimal Control Problems. *Control and Cybernetics*, **23**, 7–60.
- Piché, S.; Sayyar-Rodsari, B.; Johnson, D. and Gerules, M. (2000). Nonlinear Model Predictive Control Using Neural Networks. *IEEE Control Systems Magazine*, pages 53–62.
- Polak, E. (1973). A Historical Survey of Computational Methods in Optimal Control. *SIAM Review*, **15**, 553–584.
- Polak, E. (1997). *Optimization. Algorithms and Consistent Approximations*, volume 124 of *Applied Mathematical Sciences*. Springer, New York.

- Pontryagin, V. V.; Boltyanskii; Galka, R. and Mishchenko, E. (1962). *The Mathematical Theory of Optimal Processes*. Interscience Publishers, New York.
- Poulsen, N. K. (1995a). *Stokastisk Adaptiv Regulering. Del I*. IMM, Lyngby.
- Poulsen, N. K. (1995b). *Stokastisk Adaptiv Regulering. Del II. Stokastiske Systemer, Filtrering Og Regulering*. Institut for Matematisk Modellering, Danmarks Tekniske Universitet, Lyngby, Denmark, first edition.
- Poulsen, N. K. (1995c). *Stokastisk Adaptiv Regulering. Del V. Appendix*. IMM, Technical University of Denmark, Lyngby.
- Powell, M. J. D. (1977). A Fast Algorithm for Nonlinearly Constrained Optimization Calculations. In G. A. Watson, editor, *Numerical Analysis. Proceedings, Biennial Conference Dundee 1977*, volume 630 of *Lecture Notes in Mathematics*, pages 144–157. Springer-Verlag, Berlin.
- Powell, M. J. D. (1985a). On the Quadratic Programming Algorithm of Goldfarb and Idnani. *Mathematical Programming Study*, **25**, 46–61.
- Powell, M. J. D. (1985b). The Performance of Two Subroutines for Constrained Optimization on some Difficult Test Problems. In P. T. Boggs; R. H. Byrd and R. B. Schnabel, editors, *Numerical Optimization 1984*. SIAM, Philadelphia.
- Prett, D. M. and Garcia, C. E. (1988). *Fundamental Process Control*. Butterworths, Boston.
- Prett, D. M. and Gillette, R. D. (1979). Optimization and Constrained Multivariable Control of a Catalytic Cracking Unit. In *AIChE National Meeting*, Houston, TX.
- Prett, D. M. and Gillette, R. D. (1980). Optimization and Constrained Multivariable Control of a Catalytic Cracking Unit. In *Proceedings of the Joint Automatic Control Conference*.
- Propoi, A. I. (1963). Use of Linear Programming Methods for Synthesizing Sampled-Data Automatic Systems. *Automation and Remote Control*, **24**, 837–844.
- Propoi, A. I. (1965). The Maximum Principle for Discrete Control Systems. *Automation and Remote Control*, **26**, 1167–1177.
- Pytlak, R. (1999). *Numerical Methods for Optimal Control Problems with State Constraints*. Lecture Notes in Mathematics. Springer, New York.
- Qin, S. J. and Badgwell, T. A. (1996). An Overview of Industrial Model Predictive Control Technology. In J. C. Kantor; C. E. Garcia and B. Carnahan,

- editors, *Fifth International Conference on Chemical Process Control*, pages 232–256, Tahoe City, California. CACHE, American Institute of Chemical Engineers.
- Qin, S. J. and Badgwell, T. A. (2000). An Overview of Nonlinear Model Predictive Control Applications. In F. Allgöwer and A. Zheng, editors, *Nonlinear Model Predictive Control*, pages 369–392. Birkhäuser, Basel, Switzerland.
- Rakshit, A. and Sen, S. (1990). Sequential Rank-One/Rank-Two Updates for Quasi-Newton Differential Dynamic Programming. *Optimal Control Applications & Methods*, **11**, 95–101.
- Rao, C. V. (2000). *Moving Horizon Strategies for the Constrained Monitoring and Control of Nonlinear Discrete-Time Systems*. Ph.D. thesis, University of Wisconsin-Madison.
- Rao, C. V. and Rawlings, J. B. (1999). Steady States and Constraints in Model Predictive Control. *AIChE J.*, **45**, 1266–1278.
- Rao, C. V. and Rawlings, J. B. (2000). Nonlinear Moving Horizon State Estimation. In F. Allgöwer and A. Zheng, editors, *Nonlinear Model Predictive Control*, pages 45–69. Birkhäuser, Basel.
- Rao, C. V.; Wright, S. J. and Rawlings, J. B. (1998). Application of Interior-Point Methods to Model Predictive Control. *Journal of Optimization Theory and Applications*, **99**, 723–757.
- Ravn, H. F. (1999). *Discrete Time Optimal Control*. Technical University of Denmark, Lyngby, Denmark.
- Rawlings, J. B. (1999). Tutorial: Model Predictive Control Technology. In *Proceedings of the American Control Conference*, pages 662–676, San Diego, California.
- Rawlings, J. B. (2000). Tutorial Overview of Model Predictive Control. *IEEE Control Systems Magazine*, **20**, 38–52.
- Rawlings, J. B. and Muske, K. R. (1993). The Stability of Constrained Receding Horizon Control. *IEEE Trans. Automat. Contr.*, **38**, 1512–1516.
- Rawlings, J. B.; Meadows, E. S. and Muske, K. R. (1994). Nonlinear Model Predictive Control: A Tutorial and Survey. In *ADCHEM'94 Proceedings*, pages 185–197, Kyoto, Japan.
- Reinsel, G. C. (2003). *Elements of Multivariate Time Series Analysis*. Springer, New York.
- Reinsel, G. C. and Velu, R. P. (1998). *Multivariate Reduced-Rank Regression. Theory and Applications*, volume 136 of *Lecture Notes in Statistics*. Springer, New York.



- Richalet, J. (1993). Industrial Applications of Model Based Predictive Control. *Automatica*, **29**, 1251–1274.
- Richalet, J.; Rault, A.; Testud, J. L. and Papon, J. (1976). Algorithmic Control of Industrial Processes. In *Proceedings of the Fourth IFAC Symposium on Identification and System Parameter Estimation*, pages 1119–1167.
- Richalet, J.; Rault, A.; Testud, J. L. and Papon, J. (1978). Model Predictive Heuristic Control: Applications to Industrial Processes. *Automatica*, **14**, 413–428.
- Ricker, N. L. (1985). The Use of Quadratic Programming for Constrained Internal Model Control. *Ind. Eng. Chem. Process Des. Dev.*, **24**, 925–936.
- Ricker, N. L. (1990). Model Predictive Control with State Estimation. *Ind. Eng. Chem. Res.*, **29**, 374–382.
- Ricker, N. L.; Subrahmanian, T. and Sim, T. (1988). Case Studies of Model-Predictive Control in Pulp and Paper Production. In T. J. McAvoy; Y. Arkun and E. Zafiriou, editors, *Proceedings of the 1988 IFAC Workshop on Model Based Process Control*, pages 13–22. Pergamon Press, Oxford.
- Robertson, D. G.; Lee, J. H. and Rawlings, J. B. (1996). A Moving Horizon-Based Approach for Least-Squares Estimation. *AIChE J.*, **42**, 2209–2224.
- Rossiter, J. A. and Kouvaritakis, B. (2001). Modelling and Implicit Modelling for Predictive Control. *International Journal of Control*, **74**, 1085–1095.
- Rossiter, J. A.; Kouvaritakis, B. and Rice, M. J. (1998). A Numerically Robust State-Space Approach to Stable-Predictive Control Strategies. *Automatica*, **34**, 65–73.
- Rouhani, R. and Mehra, R. K. (1982). Model Algorithmic Control: Basic Theoretical Properties. *Automatica*, **18**, 401–414.
- Rugh, W. J. (1996). *Linear System Theory*. Prentice Hall, Upper Saddle River, New Jersey, second edition.
- Samuelson, P. A. (1966). *The Collected Scientific Papers of Paul A. Samuelson*. MIT Press, Cambridge, MA.
- Sayed, A. H. (2001). A Framework for State-Space Estimation with Uncertain Models. *IEEE Transactions on Automatic Control*, **46**, 998–1013.
- Sayed, A. H. and Kailath, T. (1994). A State-Space Approach to Adaptive RLS Filtering. *IEEE Signal Processing Magazine*, **11**, 18–60.
- Schittkowski, K. (1981a). The Nonlinear Programming Method of Wilson, Han, and Powell with an Augmented Lagrangian Type Line Search Function. Part 1: Convergence Analysis. *Numerische Mathematik*, **38**, 83–114.

- Schittkowski, K. (1981b). The Nonlinear Programming Method of Wilson, Han, and Powell with an Augmented Type Line Search Function. Part 2: An Efficient Implementation with Linear Least Squares Subproblems. *Numerische Mathematik*, **38**, 115–127.
- Schittkowski, K. (2002). *Numerical Data Fitting in Dynamical Systems*. Kluwer Academic Publishers, Dordrecht.
- Schittkowski, K. (2003). QL: A Fortran Code for Convex Quadratic Programming - User's Guide. Technical report, Department of Mathematics, University of Bayreuth.
- Schlegel, M. and Marquardt, W. (2001). Sensitivity Analysis of Linearly-Implicit Differential-Algebraic Systems by One-Step Extrapolation. Technical Report LPT-2001-30, Lehrstuhl für Prozesstechnik, RWTH Aachen.
- Schmid, C. and Biegler, L. T. (1994). Quadratic Programming Methods for Reduced Hessian SQP. *Computers and Chemical Engineering*, **18**, 817–832.
- Schwarm, A. T. and Nikolaou, M. (1999). Chance-Constrained Model Predictive Control. *AIChE Journal*, **45**, 1743–1752.
- Schwartz, A. L. (1996). *Theory and Implementation of Numerical Methods Based on Runge-Kutta Integration for Solving Optimal Control Problems*. Ph.D. thesis, University of California at Berkeley.
- Scokaert, P. O. M. and Rawlings, J. B. (1998). Constrained Linear Quadratic Regulation. *IEEE Trans. Automat. Contr.*, **43**, 1163–1169.
- Scokaert, P. O. M. and Rawlings, J. B. (1999). Feasibility Issues in Linear Model Predictive Control. *AIChE J.*, **45**, 1649–1659.
- Scokaert, P. O. M.; Rawlings, J. B. and Meadows, E. S. (1997). Discrete-Time Stability with Perturbations: Application to Model Predictive Control. *Automatica*, **33**, 463–370.
- Scokaert, P. O. M.; Mayne, D. Q. and Rawlings, J. B. (1999). Suboptimal Model Predictive Control (Feasibility Implies Stability). *IEEE Transactions on Automatic Control*, **44**, 648–654.
- Seber, G. A. F. and Wild, C. J. (1989). *Nonlinear Regression*. Wiley, New York.
- Seborg, D. E.; Edgar, T. F. and Shah, S. L. (1986). Adaptive Control Strategies for Process Control: A Survey. *AIChE J.*, **32**, 881–913.
- Seborg, D. E.; Edgar, T. F. and Mellichamp, D. A. (1989). *Process Dynamics and Control*. Wiley, New York.

- Seborg, D. E.; Edgar, T. F. and Mellichamp, D. A. (2004). *Process Dynamics and Control*. Wiley, second edition.
- Sen, S. and Yakowitz, S. J. (1987). A Quasi-Newton Differential Dynamic Programming Algorithm for Discrete-Time Optimal Control. *Automatica*, **23**, 749–752.
- Serban, R. and Petzold, L. R. (2000). COOPT - A Software Package for Optimal Control of Large-Scale Differential-Algebraic Equation Systems. Technical report, Department of Mechanical and Environmental Engineering, UCSB.
- Shinskey, F. G. (1988). *Process Control Systems. Application, Design, and Tuning*. McGraw-Hill, New York, third edition.
- Shook, D. S.; Mohtadi, C. and Shah, S. L. (1992). A Control-Relevant Identification Strategy for GPC. *IEEE Transactions on Automatic Control*, **37**, 975–980.
- Sidje, R. B. (1998). Expokit: A Software Package for Computing Matrix Exponentials. *ACM Transactions on Mathematical Software*, **24**, 130–156.
- Simon, H. A. (1956). Dynamic Programming under Uncertainty with a Quadratic Criterion Function. *Econometrica*, **24**, 74–81.
- Sjöberg, J.; Zhang, Q.; Ljung, L.; Benveniste, A.; Deylon, B.; Glorennec, P.-Y.; Hjalmarsson, H. and Juditsky, A. (1995). Nonlinear Black-Box Modeling in System Identification: A Unified Overview. *Automatica*, **31**, 1691–1724.
- Söderström, T. (2002). *Discrete-Time Stochastic Systems*. Springer, London.
- Söderström, T. and Stoica, P. (1989). *System Identification*. Prentice Hall, New York.
- Soeterboek, R. (1992). *Predictive Control - A Unified Approach*. Prentice Hall, Englewood Cliffs, NJ.
- Steinbach, M. C. (1994). A Structured Interior Point SQP Method for Nonlinear Optimal Control Problems. In R. Bulirsch and D. Kraft, editors, *Computational Optimal Control*, pages 213–222. Birkhäuser Verlag, Basel.
- Steinbach, M. C. (1995). *Fast Recursive SQP Methods for Large-Scale Optimal Control Problems*. Ph.D. thesis, University of Heidelberg, Germany.
- Stern, L. E. (1984). Criteria of Optimality in the Infinite-Time Optimal Control Problem. *Journal of Optimization Theory and Applications*, **44**, 497–508.
- Stoica, P. and Nehorai, A. (1989). On Multistep Prediction Error Methods for Time Series Models. *Journal of Forecasting*, **8**, 357–368.

- Strand, S. (2003). Use of MPC in Statoil. Presentation at the Nordic Process Control Workshop 2003, Trondheim.
- Striebel, C. (1965). Sufficient Statistics in the Optimum Control of Stochastic Systems. *J. Math. Anal. Appl.*, **12**, 576–592.
- Tabak, D. and Kuo, B. C. (1971). *Optimal Control by Mathematical Programming*. Prentice-Hall.
- Takayama, A. (1985). *Mathematical Economics*. Cambridge University Press, Cambridge, second edition.
- Tenny, M. J. (2002). *Computational Strategies for Nonlinear Model Predictive Control*. Ph.D. thesis, University of Wisconsin-Madison.
- Tenny, M. J. and Rawlings, J. B. (2002). Efficient Moving Horizon Estimation and Nonlinear Model Predictive Control. In *Proceedings of the American Control Conference*, pages 4475–4480, Anchorage.
- Tenny, M. J.; Wright, S. J. and Rawlings, J. B. (2002). Nonlinear Model Predictive Control Via Feasibility Perturbed Sequential Quadratic Programming. Technical Report Optimization Technical Report 02-06, Computer Sciences Department, University of Wisconsin-Madison.
- Thau, F. E. and Witsenhausen, H. S. (1966). A Comparison of Closed-Loop and Open-Loop Optimum Systems. *IEEE Transactions on Automatic Control*, **AC-11**, 619–621.
- Theil, H. (1957). A Note on Certainty Equivalence in Dynamic Planning. *Econometrica*, **24**, 346–349.
- Tiagounov, A. A.; Buijs, J.; Weiland, S. and De Moor, B. (2003). Long Horizon Model Predictive Control for Nonlinear Industrial Processes. In *European Control Conference ECC 2003*, University of Cambridge, UK.
- Tousain, R. L. and Bosgra, O. H. (2000). Efficient Dynamic Optimization for Nonlinear Model Predictive Control - Application to a High-Density Polyethylene Grade Change Problem. In *Proceedings of the 39th IEEE Conference on Decision and Control*, pages 760–765, Sydney, Australia. IEEE.
- Tyler, M. L. and Morari, M. (1999). Propositional Logic in Control and Monitoring Problems. *Automatica*, **35**, 565–582.
- Vada, J. (2000). *Prioritized Infeasibility Handling in Linear Model Predictive Control: Optimality and Efficiency*. Ph.D. thesis, NTNU Trondheim.
- Vada, J.; Slupphaug, O. and Johansen, T. A. (1999). Efficient Infeasibility Handling in Linear MPC Subject to Prioritized Constraints. In *European Control Conference ECC'99*, Karlsruhe, Germany. IFAC.

- Van Loan, C. F. (1978). Computing Integrals Involving the Matrix Exponential. *IEEE Transactions on Automatic Control*, **23**, 395–404.
- van Overschee, P. and de Moor, B. (1996). *Subspace Identification for Linear Systems. Theory - Implementation - Applications..* Kluwer Academic Publishers, Boston.
- Vassiliadis, V. S.; Sargent, R. W. H. and Pantelides, C. C. (1994a). Solution of a Class of Multistage Dynamic Optimization Problems. 1. Problems Without Path Constraints. *Ind. Eng. Chem. Res.*, **33**, 2111–2122.
- Vassiliadis, V. S.; Sargent, R. W. H. and Pantelides, C. C. (1994b). Solution of a Class of Multistage Dynamic Optimization Problems. 2. Problems with Path Constraints. *Ind. Eng. Chem. Res.*, **33**, 2123–2133.
- Verhaegen, M. and Van Dooren, P. (1986). Numerical Aspects of Different Kalman Filter Implementations. *IEEE Transactions on Automatic Control*, **31**, 907–917.
- Vidal, R. V. V. (1987). On the Sufficiency of the Linear Maximum Principle for Discrete-Time Control Problems. *Journal of Optimization Theory and Applications*, **54**, 583–589.
- Vidyasagar, M. (1993). *Nonlinear Systems Analysis*. Prentice Hall, Englewood Cliffs, NJ, second edition.
- Waldraff, W.; King, R. and Gilles, E. D. (1997). Optimal Feeding Strategies by Adaptive Mesh Selection for Fed-Batch Bioprocesses. *Bioprocess Engineering*, **17**, 221–227.
- Weizsäcker, C. C. V. (1965). Existence of Optimal Programs of Accumulation of an Infinite Time Horizon. *Review of Economic Studies*, **32**, 85–104.
- Wilson, R. B. (1963). *A Simplicial Algorithm for Concave Programming*. Ph.D. thesis, Graduate School of Business Administration, Harvard University.
- Witsenhausen, H. S. (1969). Inequalities for the Performance of Suboptimal Uncertain Systems. *Automatica*, **5**, 507–512.
- Witsenhausen, H. S. (1970). On Performance Bounds for Uncertain Systems. *SIAM Journal of Control*, **8**, 55–89.
- Witsenhausen, H. S. (1971). Separation of Estimation and Control for Discrete Time Systems. *Proceedings of the IEEE*, **59**, 1557–1566.
- Wonham, W. M. (1968). On the Separation Theorem of Stochastic Control. *SIAM Journal of Control*, **6**, 312–326.
- Wood, R. K. and Berry, M. W. (1973). Terminal Composition Control of a Binary Distillation Column. *Chemical Engineering Science*, **29**, 1808.

- Wright, S. J. (1993). Interior-Point Methods for Optimal Control of Discrete-Time Systems. *Journal of Optimization Theory and Applications*, **77**, 161–187.
- Wright, S. J. (1996). Applying New Optimization Algorithms to Model Predictive Control. In J. C. Kantor; C. E. Garcia and B. Carnahan, editors, *Fifth International Conference on Chemical Process Control*, pages 147–155, Tahoe City, California. CACHE, American Institute of Chemical Engineers.
- Wright, S. J. (1997). *Primal-Dual Interior-Point Methods*. SIAM, Philadelphia.
- Wright, S. J. and Tenny, M. J. (2002). A Feasible Trust-Region Sequential Quadratic Programming Algorithm. Technical Report Optimization Technical Report 02-05, Computer Sciences Department, University of Wisconsin-Madison.
- Yakowitz, S. and Rutherford, B. (1984). Computational Aspects of Discrete-Time Optimal Control. *Applied Mathematics and Computation*, **15**, 29–45.
- Yakowitz, S. J. (1986). The Stageswise Kuhn-Tucker Condition and Differential Dynamic Programming. *IEEE Transactions on Automatic Control*, **AC-31**, 25–30.
- Yakowitz, S. J. (1988). Theoretical and Computational Advances in Differential Dynamic Programming. *Control and Cybernetics*, **17**, 173–189.
- Yakowitz, S. J. (1989). Algorithms and Computational Techniques in Differential Dynamic Programming. *Control and Dynamic Systems*, **31**, 75–91.
- Ydstie, B. E. (1997). Certainty Equivalence Adaptive Control: What’s New in the Gap. In J. C. Kantor; C. E. Garcia and B. Carnahan, editors, *Chemical Process Control - V. Assessment and New Directions for Research*, volume 93 of *AIChE Symposium Series 316*, pages 9–23. CACHE, American Institute of Chemical Engineers.
- Young, P. C. (1981). Parameter Estimation for Continuous-Time Models - A Survey. *Automatica*, **17**, 23–39.
- Zadeh, L. A. and Whalen, B. H. (1962). On Optimal Control and Linear Programming. *IRE Transactions on Automatic Control*, **7**, 45.
- Zheng, A. and Morari, M. (1995). Stability of Model Predictive Control with Mixed Constraints. *IEEE Transactions of Automatic Control*, **40**, 1818–1823.
- Zhu, G.-Y. and Henson, M. A. (2002). Model Predictive Control of Interconnected Linear and Nonlinear Processes. *Ind. Eng. Chem. Res.*, **41**, 801–816.

- 
- Zhu, G.-Y.; Henson, M. A. and Ogunnaike, B. A. (2000). A Hybrid Model Predictive Control Strategy for Nonlinear Plant-Wide Control. *Journal of Process Control*, **10**, 449–458.
- Zhu, G.-Y.; Henson, M. A. and Megan, L. (2001). Dynamic Modeling and Linear Model Predictive Control of Gas Pipeline Networks. *Journal of Process Control*, **11**, 129–148.
- Zhu, Y. (2001). *Multivariable System Identification for Process Control*. Pergamon, Amsterdam.

# Vitae

John Bagterp Jørgensen was born in Herning, Denmark on May 9th, 1972.

He graduated from Herning Gymnasium in 1991 with mathematics, physics, and chemistry at level A and with the result 10.6. In the fall of 1991, he attended The Technical University of Denmark. In the summer of 1994 he won an energy research award from the Nordic Council and used that to finance a visit at the research group of professor Dr. Sigurd Skogestad in the Department of Chemical Engineering at the Norwegian Technical University in Trondheim, Norway. He received his Master of Science degree in Engineering in September 1997 from The Technical University of Denmark with the result 11.8. He began his graduate studies in the fall of 1997 when he joined the research group of professor Dr. Sten Bay Jørgensen in the Department of Chemical Engineering at The Technical University of Denmark. In the period from January to August 2000 he visited the research group of professor Dr. James B. Rawlings in the Department of Chemical Engineering at the University of Wisconsin-Madison. Since, 1998 he has attended an undergraduate program of economics and mathematical science at the University of Copenhagen, Denmark.

He has been a teaching assistant in Chemical Engineering Thermodynamics as well as in Chemical Process Control in the Department of Chemical Engineering at the Technical University of Denmark. Furthermore, he has given lectures in Chemical Process Control. In the fall of 1999, he established, gave lectures and conducted computer exercises in a model predictive control course in the Department of Chemical Engineering at the Technical University of Denmark.

He has served as process development consultant for TarcoNord A/S. Currently he is president and CEO of 2-control ApS. He also serves as a board member of the industrial consortium on model based control and monitoring in Denmark.

His mother tongue is danish and beyond that he speaks english, german and swedish. He has lived and studied in Denmark, Germany, Norway and USA. He currently lives in Sweden.